Process Rate Semantics

Representing Biochemical Systems as Collectives of Interacting Automata

Luca Cardelli

Microsoft Research
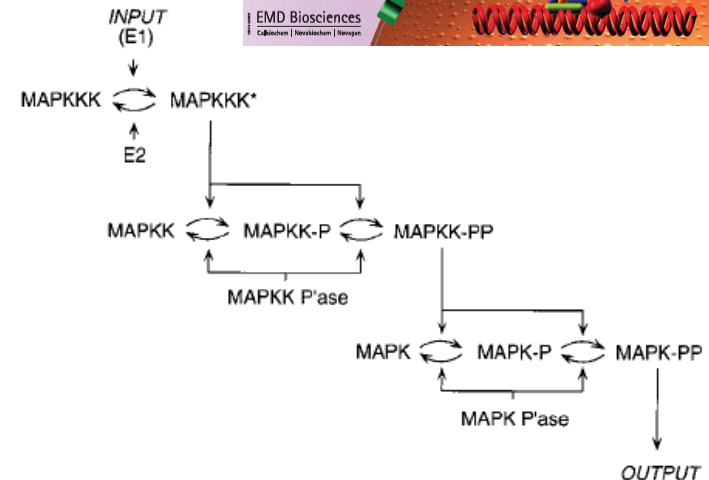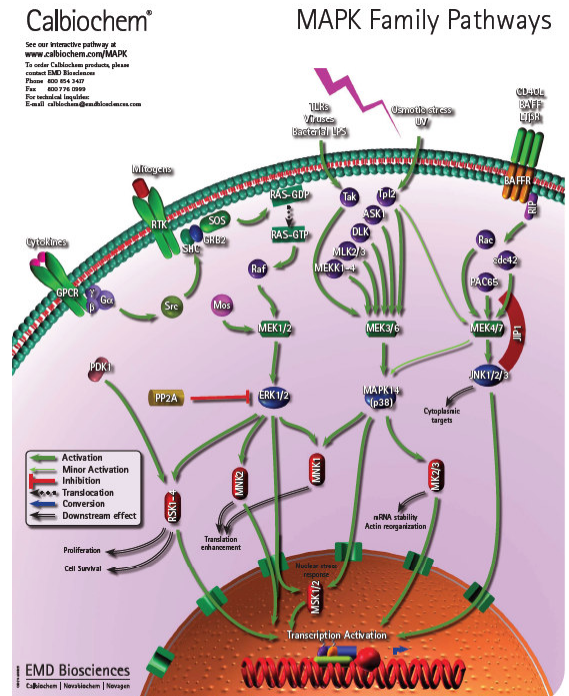
Aims

- Connections between modeling approaches
  - Connecting the discrete/concurrent/stochastic/molecular approach
  - to the continuous/sequential/deterministic/population approach

- Connecting syntax with semantics
  - Syntax = model (equations/programs/diagrams/blobs etc.)
  - Semantics = state space (generated by the syntax)
  - N.B. model ≠ state space !!
    - The same model can be interpreted in different state spaces
    - Different models can have the same state space
    - Different models of the same state space can support different analysis

- Ultimately, connections between analysis techniques
  - We need (and sometimes have) good semantic techniques to analyze state spaces (e.g. calculus, but also modelchecking)
  - But we need equally good syntactic techniques to structure complex models (e.g. compositionality) and analyze them (e.g. process algebra)
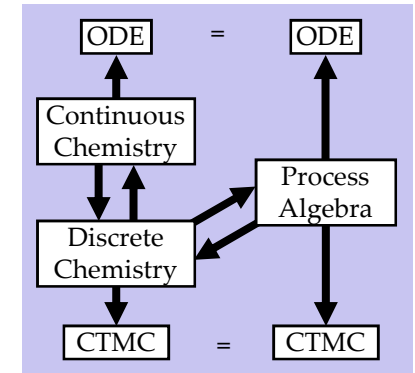
Motivation: Cells Compute

- No survival without computation!
  - Finding food
  - Avoiding predators

- *How* do they compute?
  - Unusual computational paradigms.
  - Proteins: do they work like electronic circuits?
  - Genes: what kind of software is that?

- Signaling networks
  - Clearly "information processing"
  - They are "just chemistry": molecule interactions
  - But what are their principles and algorithms?

- Complex, higher-order interactions
  - MAPKKK = MAP Kinase Kinase Kinase:
    that which operates on that which operates on that which operates on protein.

- General models of biological computation
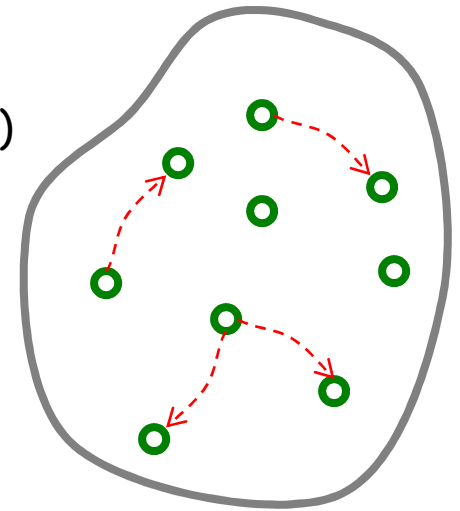 - What are the appropriate ones?

Calbiochem®   MAPK Family Pathways

EMD Biosciences

INPUT (E1) → MAPKKK ⇄ MAPKKK*
E2
MAPKK ⇄ MAPKK-P ⇄ MAPKK-PP
MAPKK P'ase
MAPK ⇄ MAPK-P ⇄ MAPK-PP
MAPK P'ase
OUTPUT

Ultrasensitivity in the mitogen-activated protein cascade, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.

(Macro)Molecules as Interacting Automata

- Concurrent   (math is based on processes, not functions)
- Asynchronous   (no global clock)
- Stochastic   (or nondeterministic)
- Stateful   (e.g. phosphorylation state)
- Discrete   (transitions between states)
- Interacting   (an "interaction" can be pretty much anything you want that changes molecular state)

- Based on work on process algebra and biological modeling; see references in related papers.

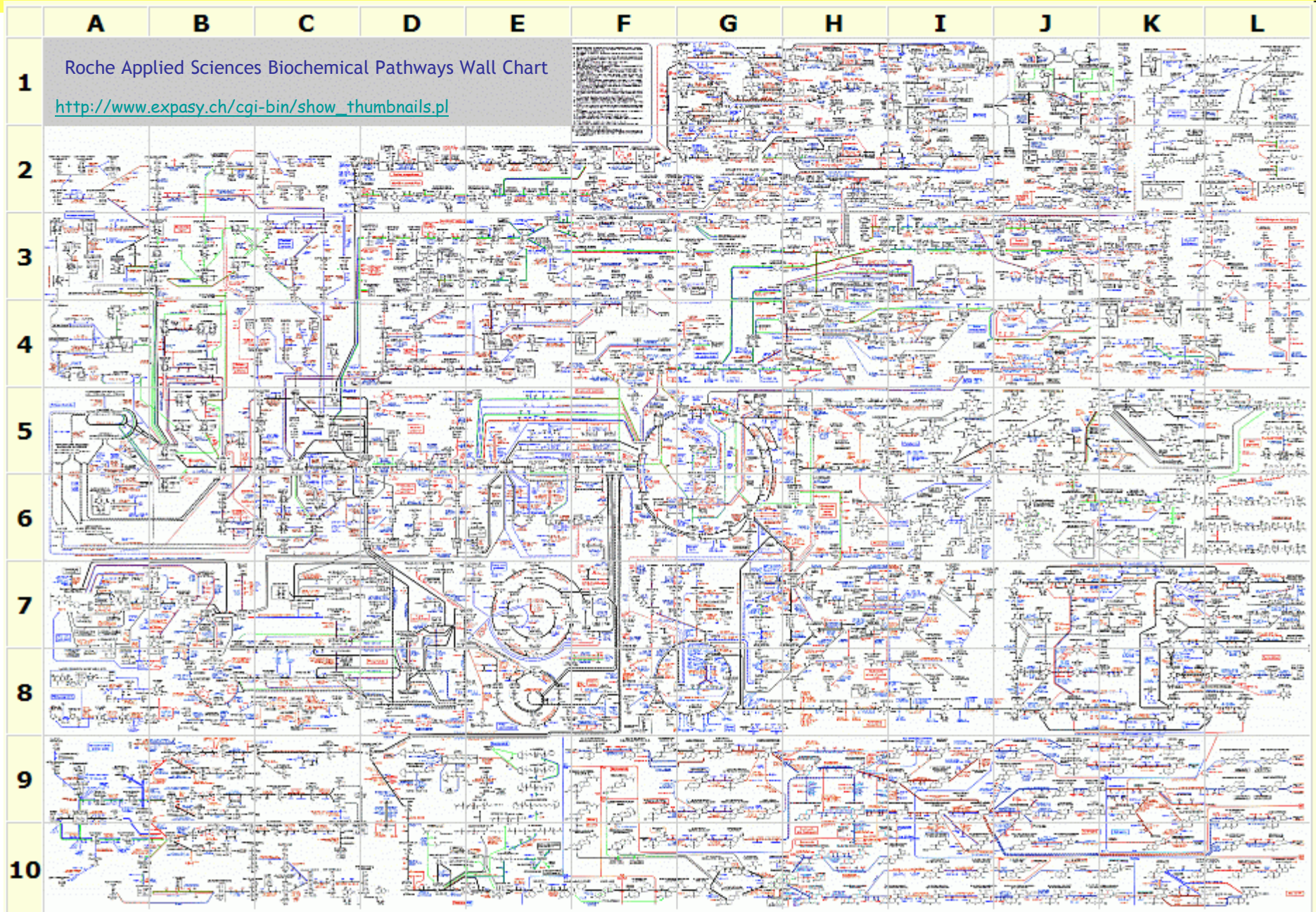# Stochastic Automata Collectives

- "Collective":
  - A large set of interacting finite state automata:
    - Not quite language automata ("large set")
    - Not quite cellular automata ("interacting" but not on a grid)
    - Not quite process algebra ("collective behavior")
    - Cf. multi-agent systems and swarm intelligence

- "Stochastic":
  - Interactions have *rates*
    - Not quite discrete (hundreds or thousands of components)
    - Not quite continuous (non-trivial stochastic effects)
    - Not quite hybrid (no "switching" between regimes)

- Very much like biochemistry
  - Which is a large set of stochastically interacting molecules/proteins
  - Are proteins finite state and subject to automata-like transitions?
    - Let's say they are, at least because:
    - Much of the knowledge being accumulated in Systems Biology is described as state transition diagrams [Kitano].
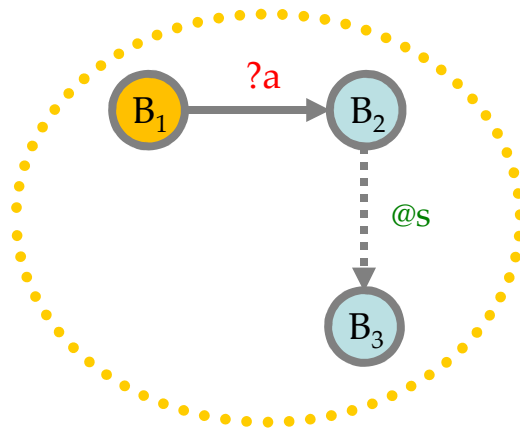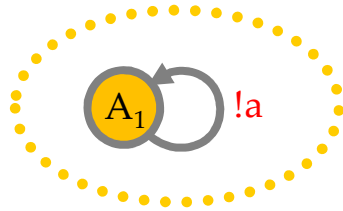
Towards <u>Systems Biology</u>

Epidermal Growth Factor Receptor Pathway Map

ErbB family

GPCR-mediated transactivation

Syntax

State Transitions!

degradation

recycling

endocytosis

small GTPase

PIP signaling

MAPK cascade

Ca signaling

transcription

cell cycle

LEGENDS

Compositionality (NOT!)

Roche Applied Sciences Biochemical Pathways Wall Chart

http://www.expasy.ch/cgi-bin/show_thumbnails.pl

Interacting Automata

- 🟠 Current State
- ┈┈▶ Decay
- ──▶ Transition
- ┈┈▶ Interaction

A_1 is a *state*

a   is a *channel* i.e. a named
interaction interface
(e.g. a surface patch)

?,!   indicate any *complementarity* of
interaction (e.g. charge)

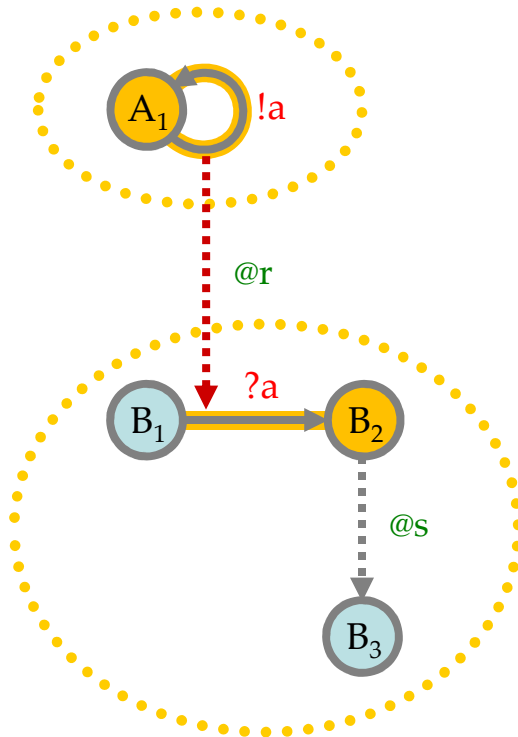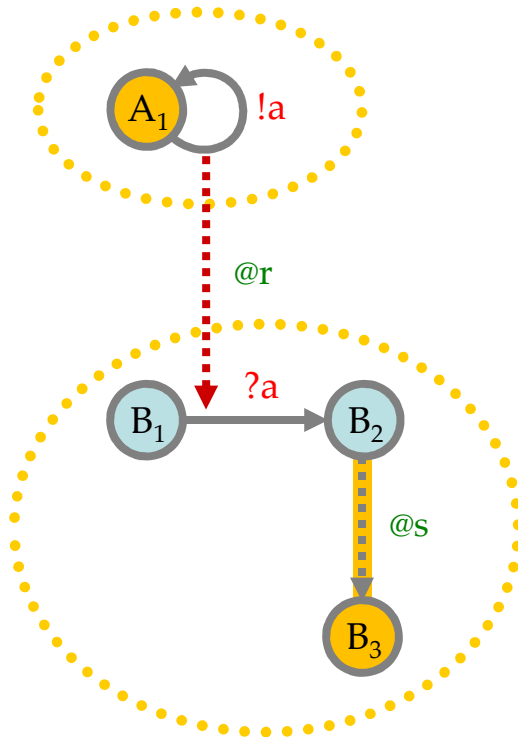?a, !a   indicate *complementary actions,*

@r, @s   are rates

Interacting Automata

Current State
Decay
Transition
Interaction

A_1 !a

Kinetic laws:

Two complementary actions may result in an interaction.

@r

?a

B_1 B_2

@s

B_3

A_1 is a *state*

a   is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)

?,!   indicate any *complementarity* of interaction (e.g. charge)

?a, !a   indicate *complementary actions,* joined by an interaction arrow

@r, @s   are rates

2008-03-13

Interacting Automata

Legend:
- ● Current State
- ┄► Decay
- ──► Transition
- ┅► Interaction

Kinetic laws:

Two complementary actions may result in an interaction.

A decay may happen spontaneously.

A_1 is a *state*

a   is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)

?,!   indicate any *complementarity* of interaction (e.g. charge)

?a, !a   indicate *complementary actions,* joined by an interaction arrow ┅►

@r, @s   are rates

The equivalent process algebra model

new a@r_1
new b@r_2
new c@r_3 ⎫ Communication channels

A_1 = ?a; A_2
A_2 = !c; A_3
A_3 = τ@λ_5; A_1

B_1 = τ@λ_2; B_2 + !a; B_3
B_2 = τ@λ_1; B_1
B_3 = ?b; B_2

C_1 = !b; C_2 + ?c; C_3
C_2 = τ@λ_3; C_1
C_3 = τ@λ_4; C_2

Automata

A_1 | B_1 | C_1 ⎫ The system and initial state

Current State
Delay
Transition
Interaction

Interactions have rates. Actions DO NOT have rates.

Interactions in a Population

Suppose this is the next interaction (stochastically chosen)

!a

A

?a   ?b

B

!b

!a

A

?a   ?b

B

!b

!a

A

?a   ?b

B

!b

!a

A

?a   ?b

B

!b

One lonely automaton cannot interact
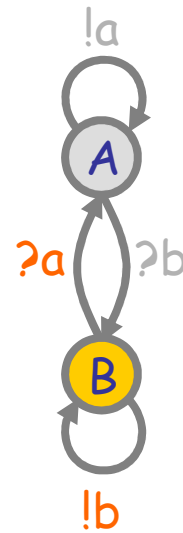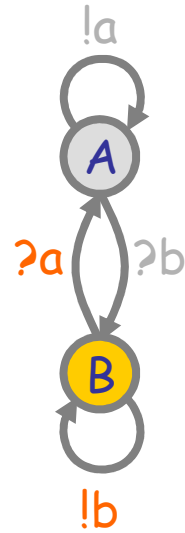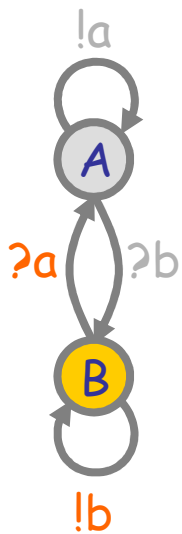
Interactions in a Population
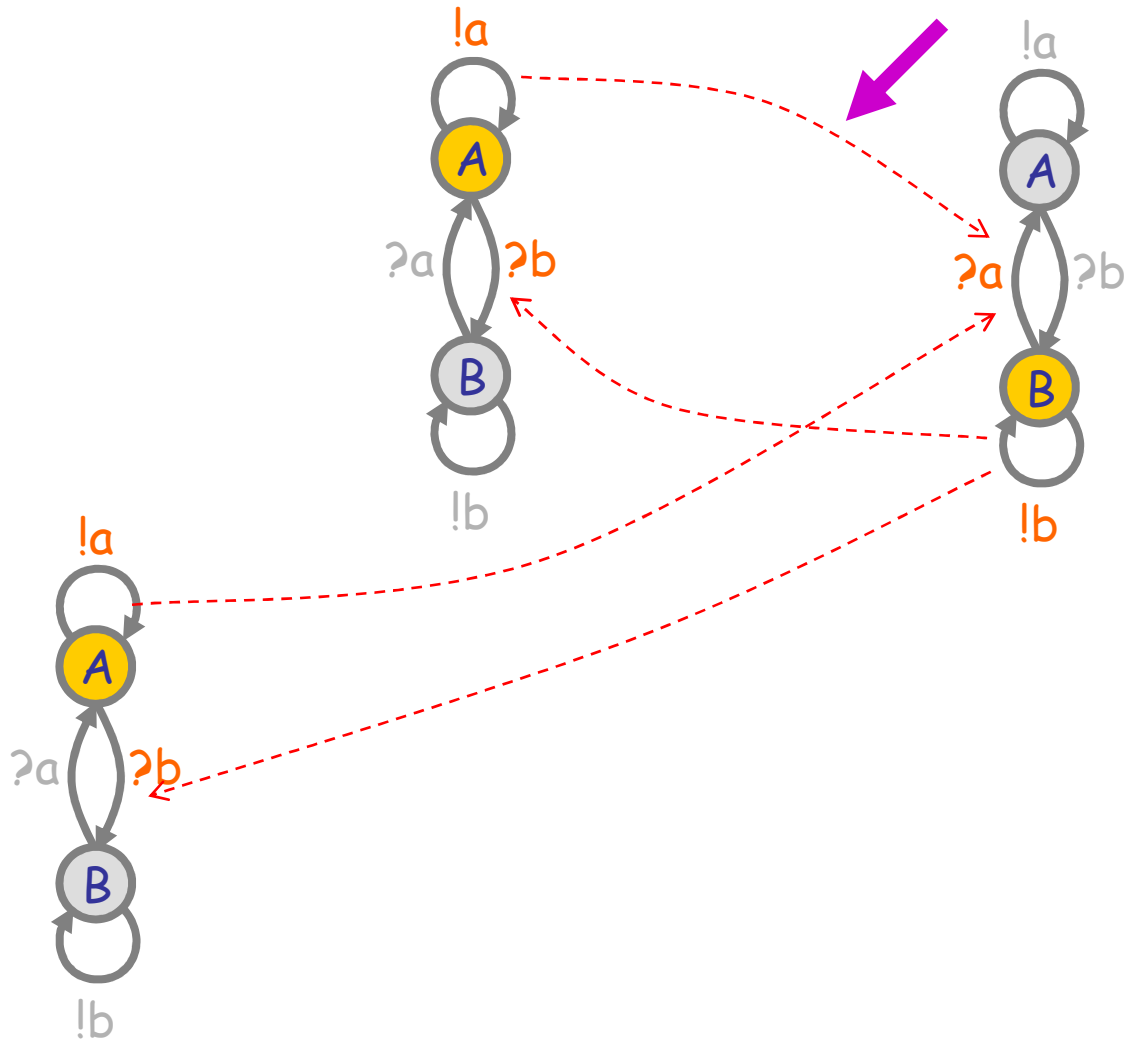
Interactions in a Population

!a

A

?a   ?b

B

!b

!a

A

?a   ?b

B

!b

✗

!a

A

?a   ?b

B

!b
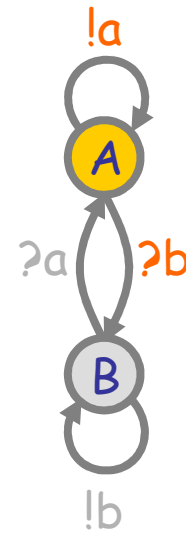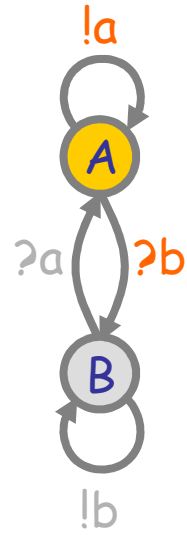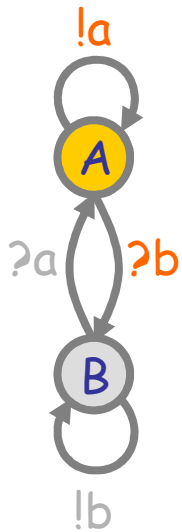
All-A stable
population

!a

A

?a ?b

B

!b

!a

A

?a ?b

B

!b

!a

A

?a ?b

B

!b

Suppose this is the
next interaction
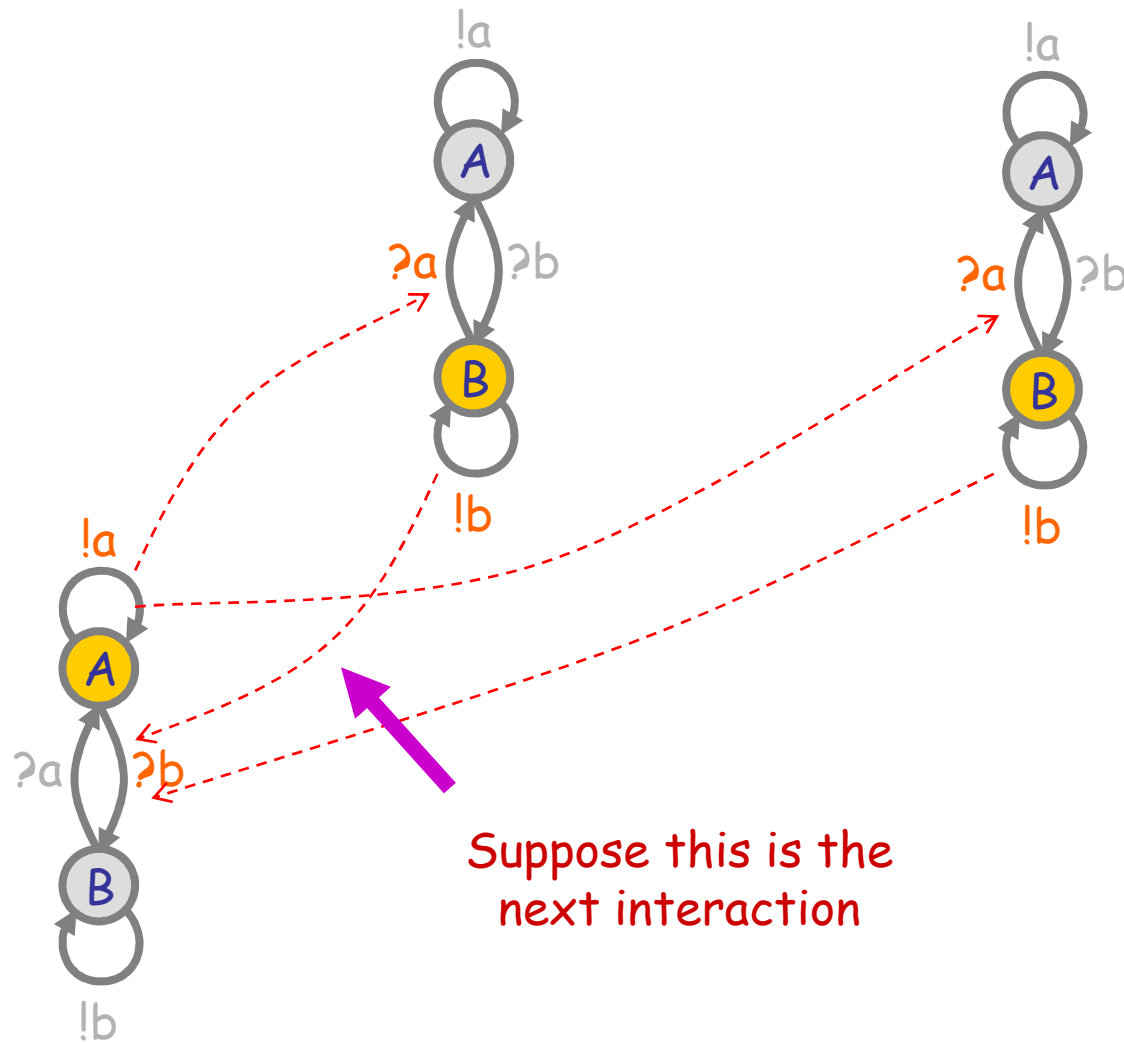
All-B stable
population

Nondeterministic
population behavior
("multistability")

CTMC Semantics

!a

A

?a   ?b

B

!b

!a

A

?a   ?b

B

!b
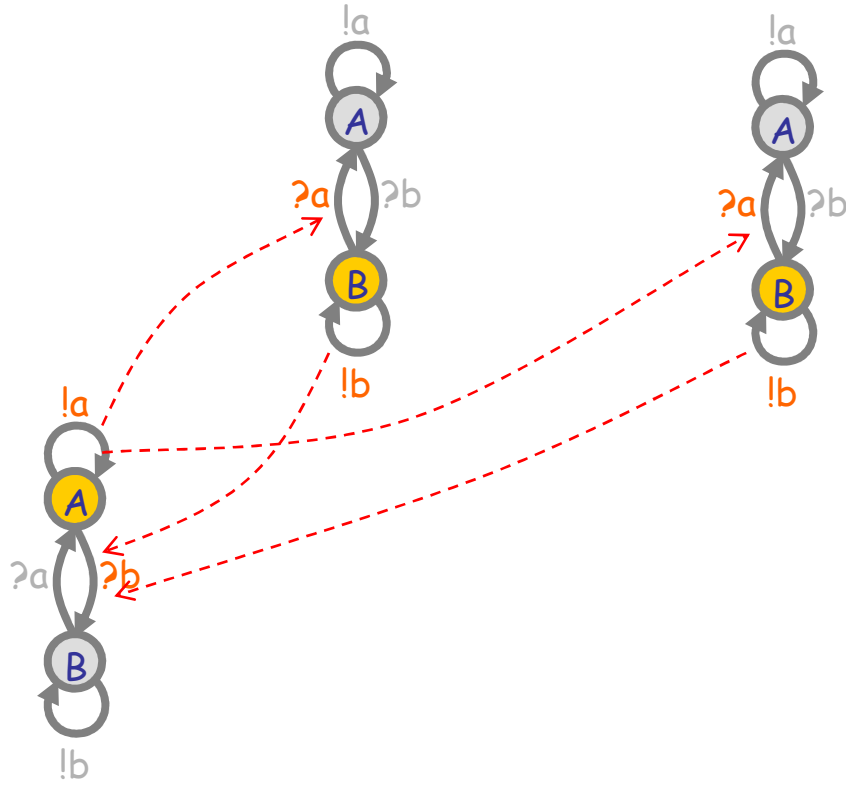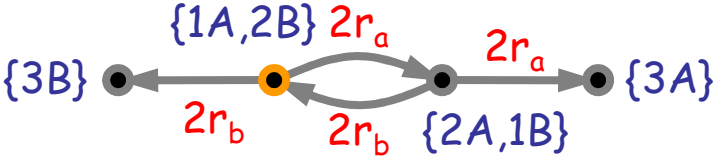
CTMC
(homogeneous) Continuous Time
Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

$$A \xrightarrow{\;r\;} B$$

Probability of holding in state A:

$$Pr(H_A > t) = e^{-rt}$$

in general, $Pr(H_A > t) = e^{-Rt}$ where R is
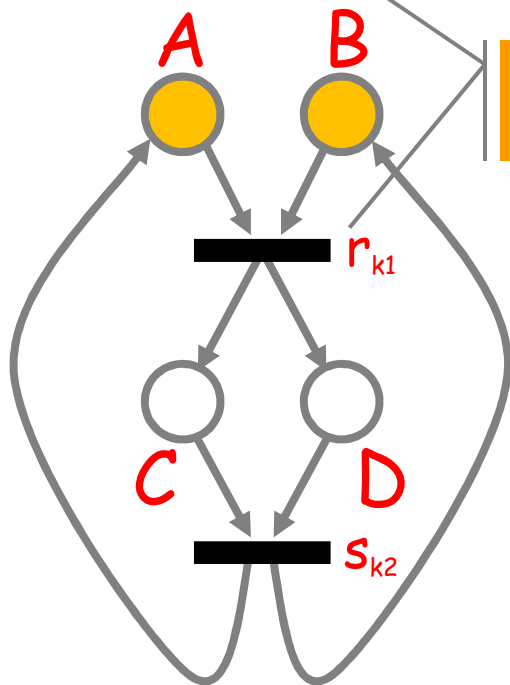the sum of all the exit rates from A

!a

A

?a   ?b

B

!b

$$\{3B\} \longleftarrow \{1A,2B\} \; 2r_a \quad 2r_a \quad \{3A\}$$

$$2r_b \qquad 2r_b \quad \{2A,1B\}$$

CTMC

Chemistry vs. Automata

A process algebra (chemistry)

A different process algebra (automata)

$$r: A + B \rightarrow_{k1} C + D$$
$$s: C + D \rightarrow_{k2} A + B$$
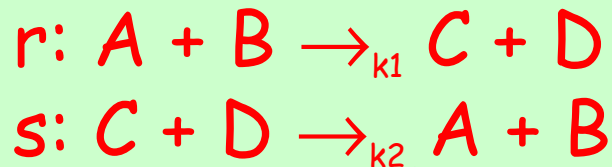
Does A become C or D?

Reaction oriented

1 line per reaction

A B

r_{k1}

C D

s_{k2}

A

B

$?s_{k2}$ $!r_{k1}$ $?r_{k1}$ $!s_{k2}$

C   D
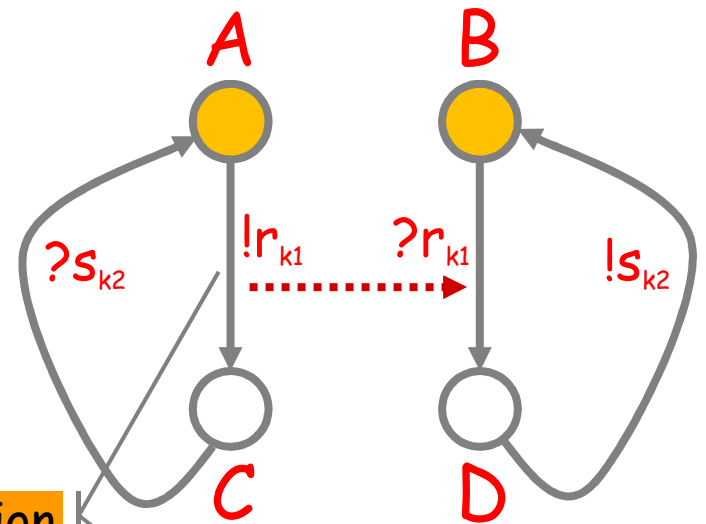
Interaction oriented

1 line per component

The same "model"

Maps to a CTMC   Maps to a CTMC

$$A = !r_{k1}; C$$
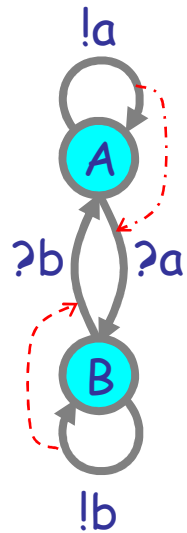$$C = ?s_{k2}; A$$
$$B = ?r_{k1}; D$$
$$D = !s_{k2}; B$$

A becomes C not D!

A Petri-Net-like representation. Precise and dynamic, but not modular, scalable, or maintainable.

A compositional graphical representation (precise, dynamic *and* modular) and the corresponding calculus.

Groupies and Celebrities

Groupies and Celebrities

Celebrity
(does not want to be like somebody else)

!a

A

?b   ?a

B

!b

```
directive sample 1.0 1000
directive plot A(); B()

new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?a; B()
and B() = do !b; B() or ?b; A()

run 100 of (A() | B())
```
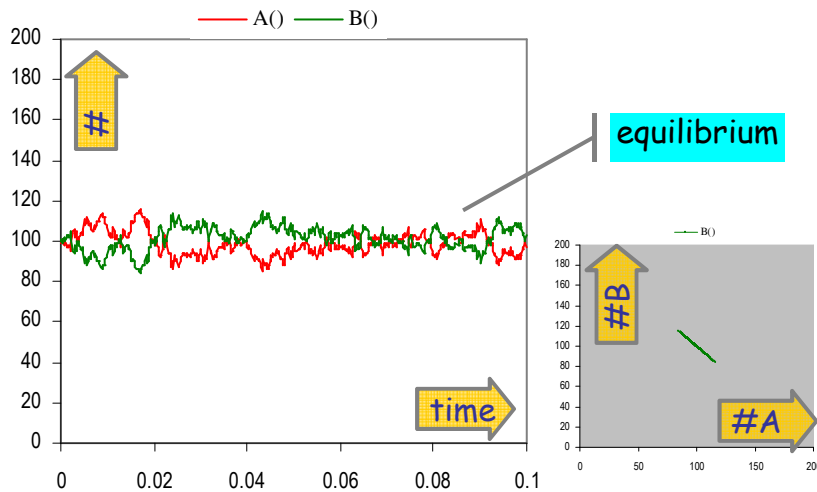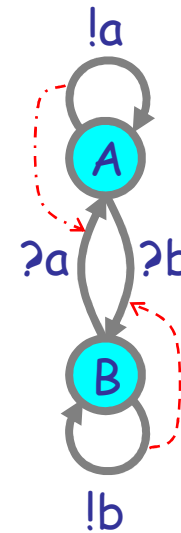
a@1.0

b@1.0

Groupie
(wants to be like somebody different)

!a

A

?a   ?b

B

!b

```
directive sample 1.0 1000
directive plot A(); B()

new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?b; B()
and B() = do !b; B() or ?a; A()

run 100 of (A() | B())
```
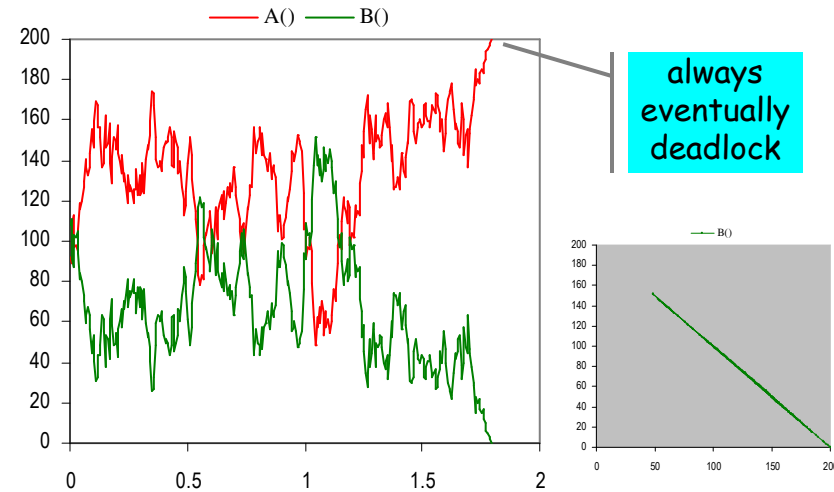
a@1.0

b@1.0

A stochastic collective of celebrities:

equilibrium

#

#B

time   #A

A stochastic collective of groupies:

always
eventually
deadlock

Stable because as soon as a A finds itself in the majority, it is more likely to
find somebody in the same state, and hence change, so the majority is weakened.

Unstable because within an A majority, an A has difficulty finding a B to
emulate, but the few B's have plenty of A's to emulate, so the majority may
switch to B. Leads to deadlock when everybody is in the same state and there is
nobody different to emulate.

Both Together

A way to break the deadlocks: Groupies with just a few Celebrities

!a   !a

Many   A few
Groupies   Celebrities

?a   ?b   ?b   ?a

!b   !b
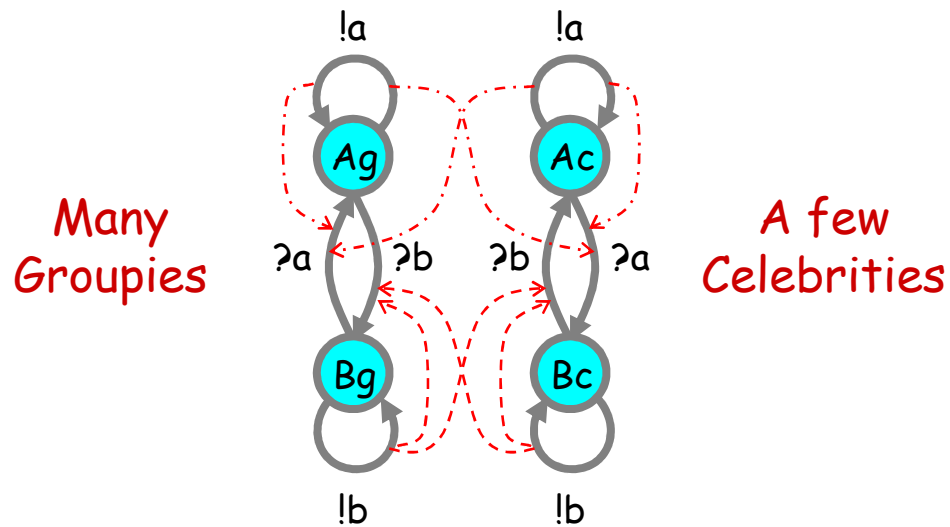
directive sample 10.0
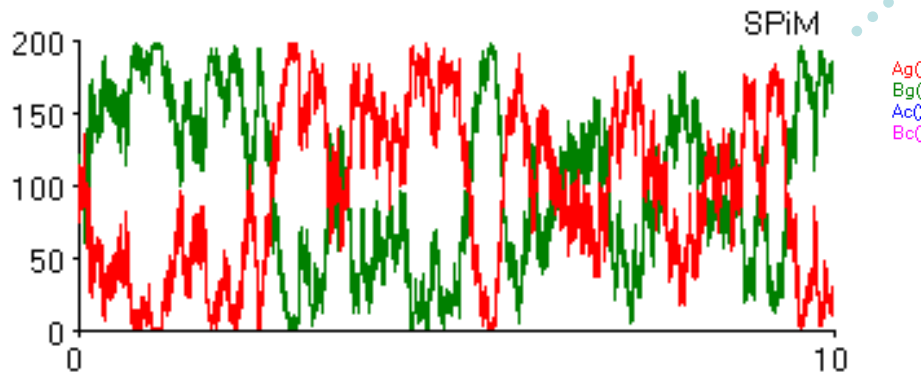directive plot Ag(); Bg(); Ac(); Bc()

new a@1.0:chan()
new b@1.0:chan()

let Ac() = do !a; Ac() or ?a; Bc()
and Bc() = do !b; Bc() or ?b; Ac()

let Ag() = do !a; Ag() or ?b; Bg()
and Bg() = do !b; Bg() or ?a; Ag()

run 1 of Ac()
run 100 of (Ag() | Bg())

never
deadlock

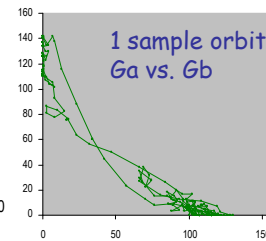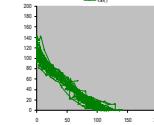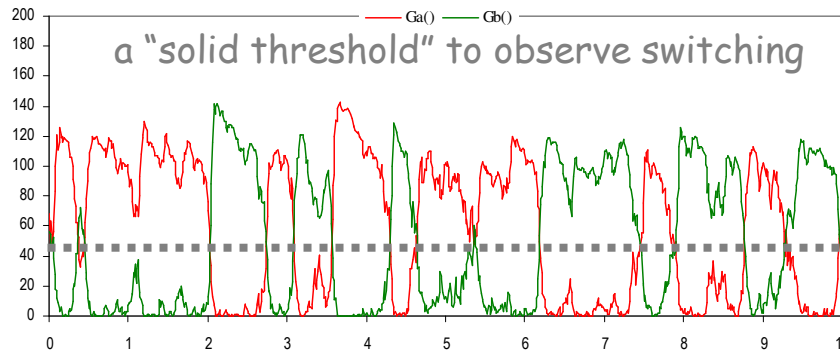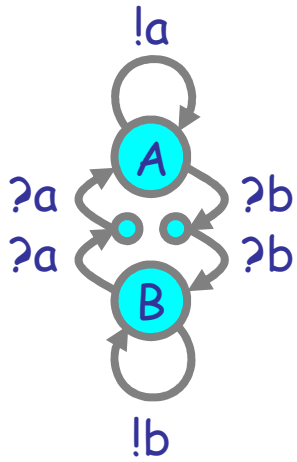SPiM

200

150

100

50

0

0   10

Ag()
Bg()
Ac()
Bc()

A tiny bit of
"noise" can make a
huge difference

Hysteric Groupies

We can get more regular behavior from groupies if they "need more convincing", or "**hysteresis**" (history-dependence), to switch states.

!a

?a   ?b
?a   ?b

A
B

!b

a "solid threshold" to observe switching

Ga()   Gb()

1 sample orbit
Ga vs. Gb

```
directive sample 10.0 1000
directive plot Ga(); Gb()

new a@1.0:chan()
new b@1.0:chan()

let Ga() = do !a; Ga() or ?b; ?b; Gb()
and Gb() = do !b; Gb() or ?a; ?a; Ga()

let Da() = !a; Da()
and Db() = !b; Db()

run 100 of (Ga() | Gb())
run   1 of (Da() | Db())
```
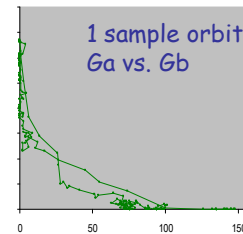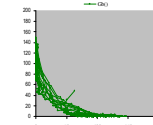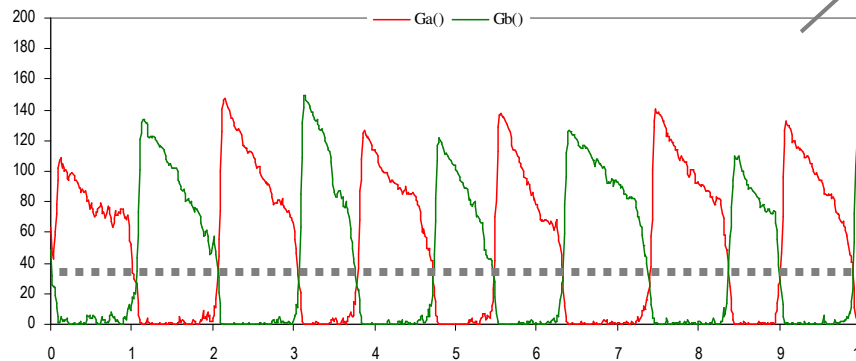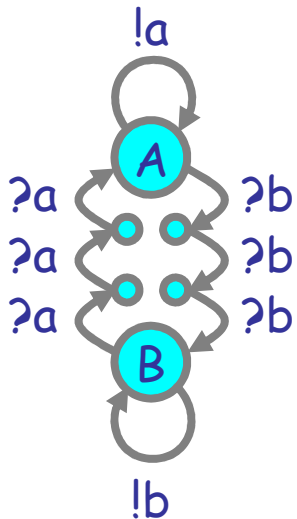
!a   !b

(With doping to break deadlocks)

N.B.: It will not oscillate without doping (noise)

"regular" oscillation

!a

?a   ?b
?a   ?b
?a   ?b

A
B

!b

Ga()   Gb()

1 sample orbit
Ga vs. Gb

```
directive sample 10.0 1000
directive plot Ga(); Gb()

new a@1.0:chan()
new b@1.0:chan()

let Ga() = do !a; Ga() or ?b; ?b; ?b; Gb()
and Gb() = do !b; Gb() or ?a; ?a; ?a; Ga()

let Da() = !a; Da()
and Db() = !b; Db()

run 100 of (Ga() | Gb())
run   1 of (Da() | Db())
```
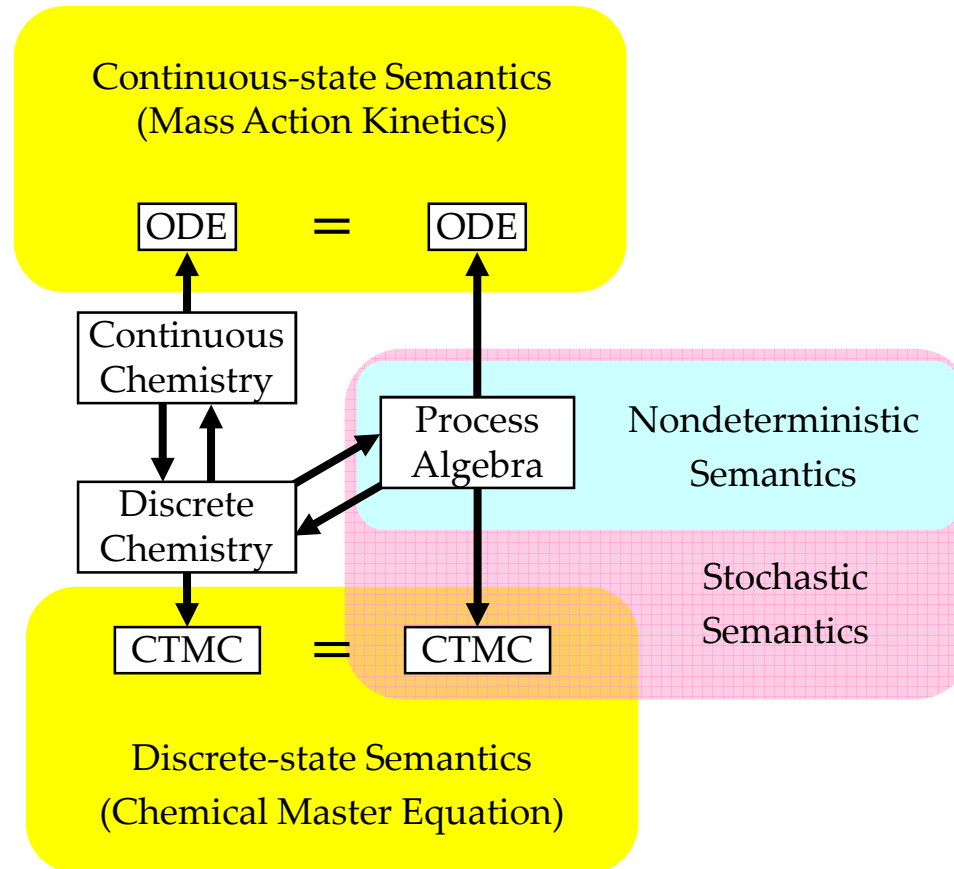
Semantics of Collective Behavior

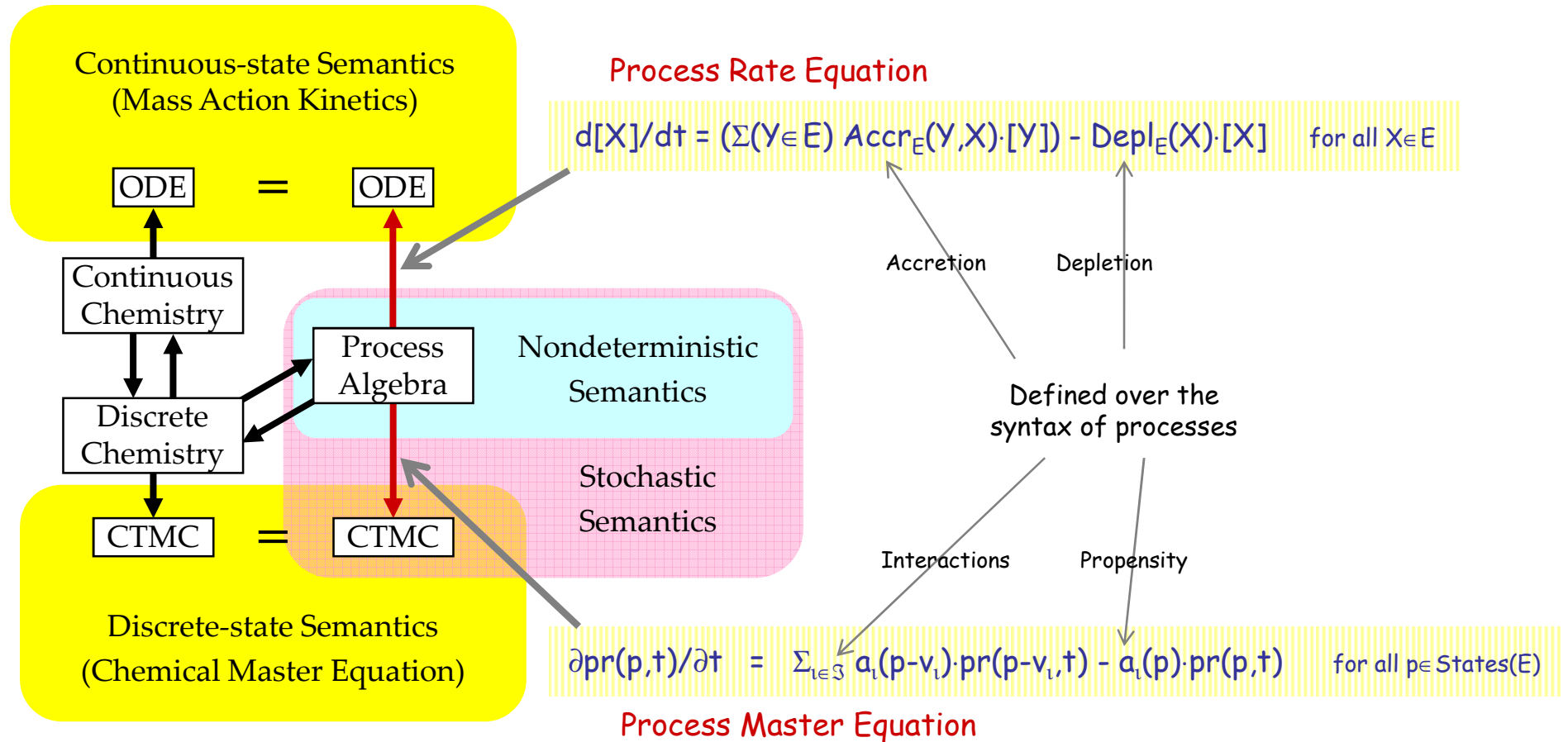# The Two Semantic Sides of Chemistry
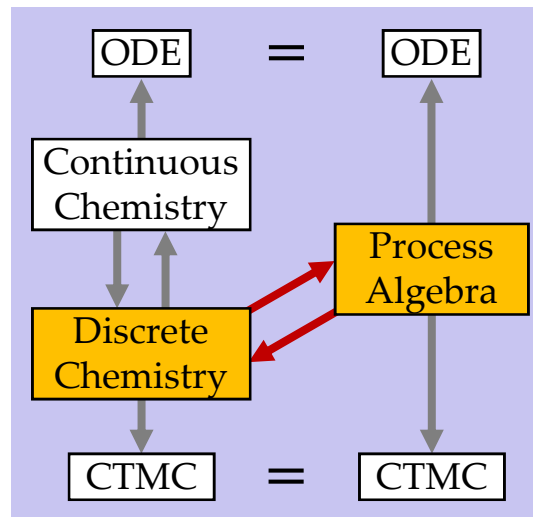

These diagrams commute via appropriate maps.

L. Cardelli: "On Process Rate Semantics" (TCS)

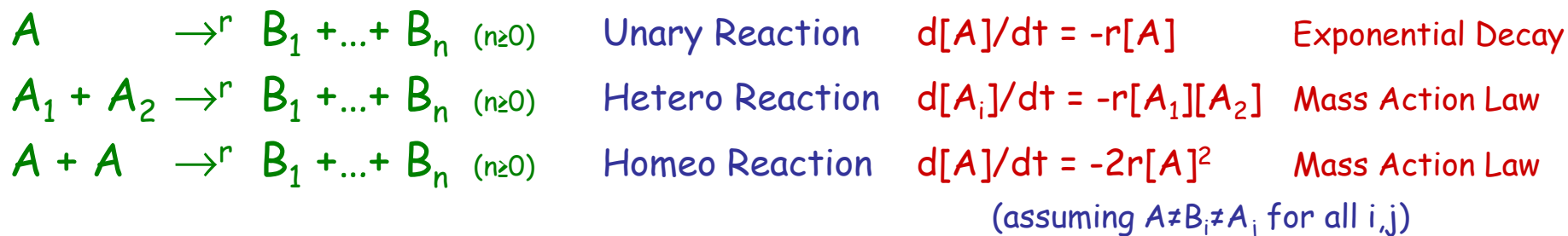L. Cardelli: "A Process Algebra Master Equation" (QEST'07)

Quantitative Process Semantics

Continuous-state Semantics
(Mass Action Kinetics)

ODE $\quad = \quad$ ODE

Continuous Chemistry

Discrete Chemistry

Process Algebra

Nondeterministic Semantics

Stochastic Semantics

CTMC $\quad = \quad$ CTMC

Discrete-state Semantics
(Chemical Master Equation)

Process Rate Equation

$$d[X]/dt = (\Sigma(Y \in E)\ Accr_E(Y,X) \cdot [Y]) - Depl_E(X) \cdot [X] \qquad \text{for all } X \in E$$

Accretion $\qquad$ Depletion

Defined over the
syntax of processes

Interactions $\qquad$ Propensity

$$\partial pr(p,t)/\partial t \;=\; \Sigma_{\iota \in \mathfrak{I}}\ a_\iota(p-v_\iota) \cdot pr(p-v_\iota,t) - a_\iota(p) \cdot pr(p,t) \qquad \text{for all } p \in States(E)$$

Process Master Equation

Stochastic Processes & Discrete Chemistry

Chemical Reactions

$A \to^r B_1 + ... + B_n$ (n≥0)   Unary Reaction $d[A]/dt = -r[A]$ Exponential Decay

$A_1 + A_2 \to^r B_1 + ... + B_n$ (n≥0)   Hetero Reaction $d[A_i]/dt = -r[A_1][A_2]$ Mass Action Law

$A + A \to^r B_1 + ... + B_n$ (n≥0)   Homeo Reaction $d[A]/dt = -2r[A]^2$ Mass Action Law

(assuming $A \neq B_i \neq A_j$ for all i,j)

No other reactions!

JOURNAL OF CHEMICAL PHYSICS   VOLUME 113, NUMBER 1

The chemical Langevin equation

Daniel T. Gillespie[a]
Research Department, Code 4T4100D, Naval Air Warfare Center, China Lake, California 93555

Genuinely *trimolecular* reactions do not physically occur in dilute fluids with any appreciable frequency. *Apparently* trimolecular reactions in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction, and involve an additional short-lived species.

Chapter IV: Chemical Kinetics
[David A. Reckhow , CEE 572 Course]

... reactions may be either elementary or non-elementary. Elementary reactions are those reactions that occur exactly as they are written, without any intermediate steps. These reactions almost always involve just one or two reactants. ... Non-elementary reactions involve a series of two or more elementary reactions. Many complex environmental reactions are non-elementary. In general, reactions with an overall reaction order greater than two, or reactions with some non-integer reaction order are non-elementary.

THE COLLISION THEORY OF REACTION RATES
www.chemguide.co.uk

The chances of all this happening if your reaction needed a collision involving more than 2 particles are remote. All three (or more) particles would have to arrive at exactly the same point in space at the same time, with everything lined up exactly right, and having enough energy to react. That's not likely to happen very often!

Trimolecular reactions:

$A + B + C \to^r D$

the measured "r" is an (imperfect) aggregate of e.g.:

$A + B \leftrightarrow AB$

$AB + C \to D$

Enzymatic reactions:

$S \xrightarrow{E} {}^r P$

the "r" is given by Michaelis-Menten (approximated steady-state) laws:

$E + S \leftrightarrow ES$

$ES \to P + E$

Reactions have rates. Molecules *do not* have rates.

Chemical Ground Form (CGF)

$E ::= 0 ~\vdots~ X=M, E$ **Reagents**

$M ::= 0 ~\vdots~ \pi;P \oplus M$ **Molecules**

$P ::= 0 ~\vdots~ X \mid P$ **Solutions**

$\pi ::= \tau_{(r)} ~\vdots~ ?a_{(r)} ~\vdots~ !a_{(r)}$ **Actions (delay, input, output)**
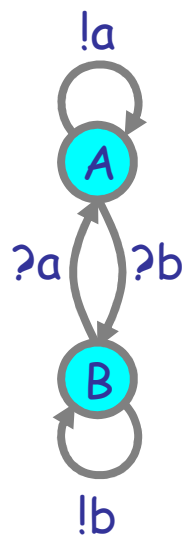
$CGF ::= E,P$ **Reagents plus Initial Conditions**

A stochastic subset of CCS
(no values, no restriction)

Interacting Automata
+ dynamic forking

(To translate chemistry to processes we need a bit more than interacting automata: we may have "+" on the right of $\rightarrow$, that is we may need "|" after π.)

$\oplus$ is stochastic choice (vs. + for chemical reactions)
0 is the null solution (P|0 = 0|P = P)
   and null molecule (M$\oplus$0 = 0$\oplus$M = M)
Each X in E is a distinct *species*
Each name a is assigned a fixed rate r: $a_{(r)}$

!a

A

?a   ?b

B

!b

Ex: Interacting Automata
(= finite-control CGFs: they use "|" only in initial conditions):

$A = !a;A \oplus ?b;B$ — Automaton in state A
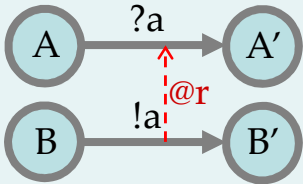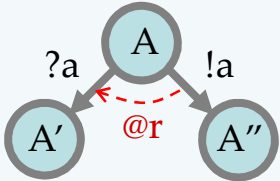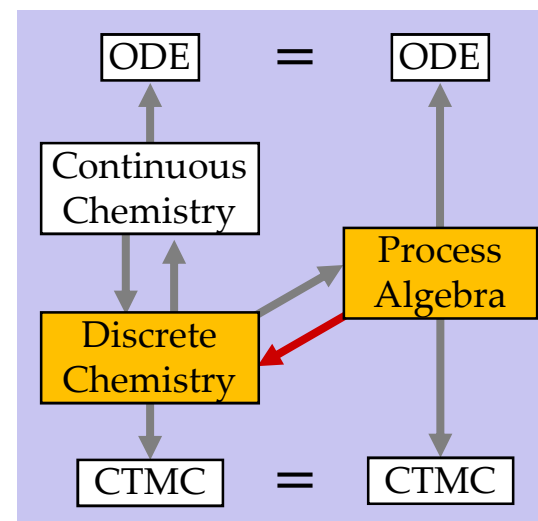
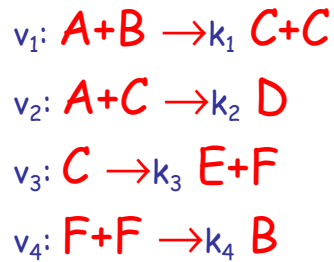$B = !b;B \oplus ?a;A$ — Automaton in state B

$A|A|B|B$ — Initial conditions: 2A and 2B

From Reagents to Reactions (by example)

Interacting Automata →	Discrete Chemistry
initial states $A \mid A \mid ... \mid A$	initial quantities $\#A_0$
	$A \dashrightarrow^r A'$
	$A+B \dashrightarrow^r A'+B'$
	$A+A \dashrightarrow^{2r} A'+A''$

From Reactions to Reagents (by example)

v_1: $A+B \rightarrow_{k_1} C+C$

v_2: $A+C \rightarrow_{k_2} D$

v_3: $C \rightarrow_{k_3} E+F$

v_4: $F+F \rightarrow_{k_4} B$

Interaction Matrix

channels and rates (1 per reaction)

Half-rate for homeo reactions

definitions (1 per species)

		$v_{1(k1)}$	$v_{2(k2)}$	$v_{3(k3)}$	$v_{4(k4/2)}$
	A	?;(C|C)	?;D		
	B	!;0			
	C		!;0	τ;(E|F)	
	D				
	E				
	F				?;B !;0
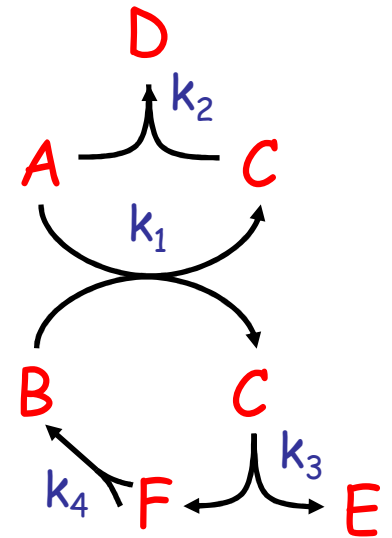
1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow_{k_i} P_i$
  add τ;P_i to <X,v_i>.

Hetero reaction v_i: $X+Y \rightarrow_{k_i} P_i$
  add ?;P_i to <X,v_i> and !;0 to <Y,v_i>

Homeo reaction v_i: $X+X \rightarrow_{k_i} P_i$
  add ?;P_i and !;0 to <X,v_i>

2: Read the result by rows:

$A = ?v_{1(k1)};(C|C) \oplus ?v_{2(k2)};D$

$B = !v_{1(k1)};0$

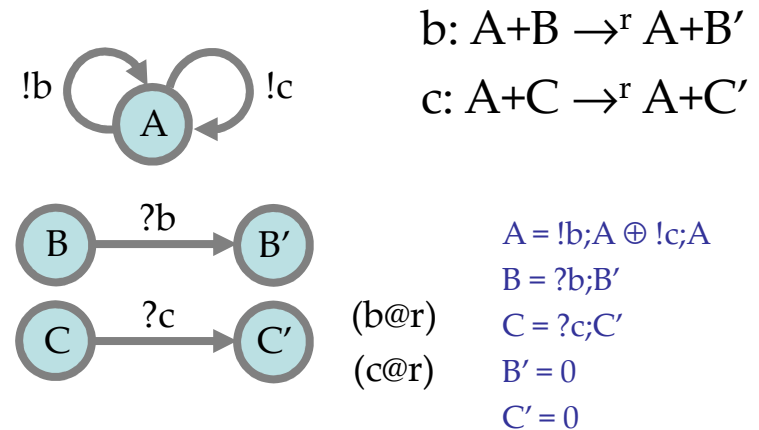$C = !v_{2(k2)};0 \oplus \tau_{k3};(E|F)$

$D = 0$

$E = 0$

$F = ?v_{4(k4/2)};B \oplus !v_{4(k4/2)};0$

D

k_2

A ——— C

k_1

B   C

k_4 F   E k_3

ODE   =   ODE

Continuous Chemistry

Process Algebra

Discrete Chemistry

CTMC   =   CTMC

Entangled vs Detangled

a: A+B $\to^r$ A+B′

a: A+C $\to^r$ A+C′

!a

A = !a;A
B = ?a;B′
C = ?a;C′
B′ = 0
C′ = 0

(a@r)

?a

?a
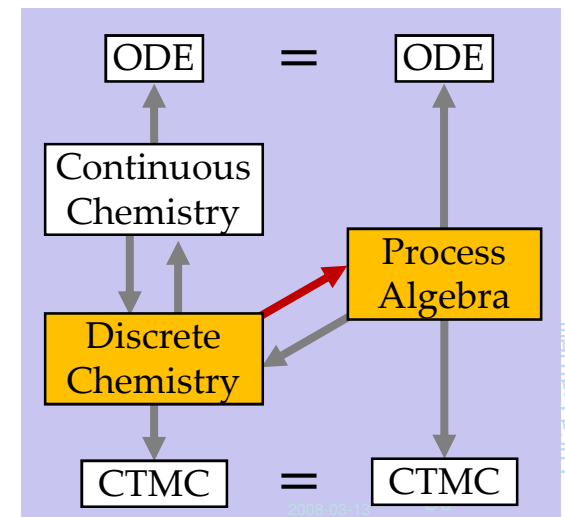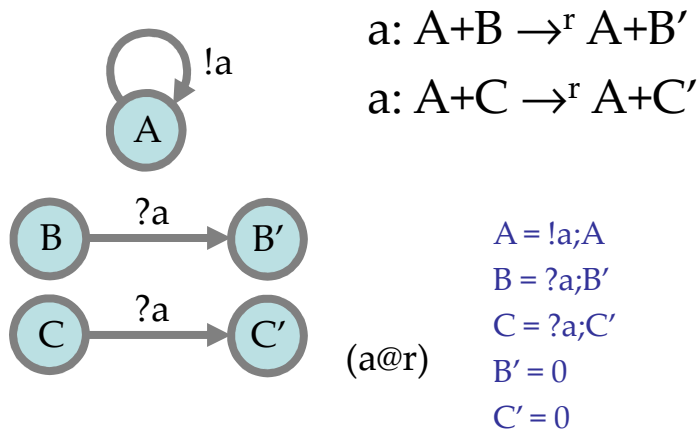
Entangled: Two reactions
on one channel

---

b: A+B $\to^r$ A+B′

c: A+C $\to^r$ A+C′

!b   !c

A = !b;A $\oplus$!c;A
B = ?b;B′
C = ?c;C′
B′ = 0
C′ = 0

(b@r)
(c@r)

?b

?c

Detangled: Two reactions
on two separate channels

We need a semantics of automata that identifies
automata that have the "same chemistry".
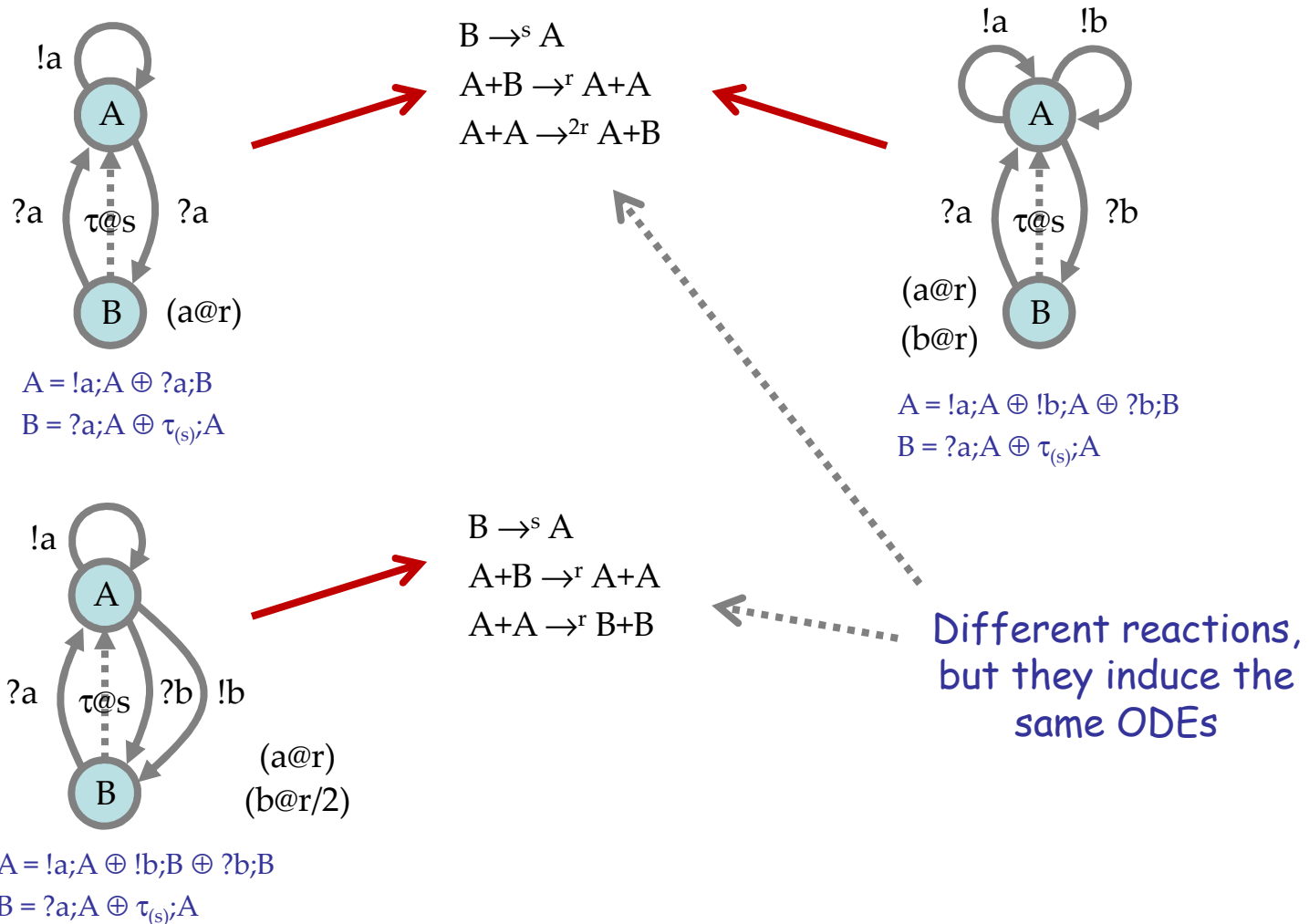
No process algebra equivalence is like this!

Entangled automata lead to more compact models than in chemistry.

Detangled automata are in simple correspondence with chemistry.

2008-03-13   31

Same Semantics

Could chemistry itself be that semantics?
No: different sets of reactions can have the same behavior!

$B \to^s A$
$A+B \to^r A+A$
$A+A \to^{2r} A+B$

!a

?a $\tau@s$?a

B (a@r)

$A = !a;A \oplus ?a;B$
$B = ?a;A \oplus \tau_{(s)};A$

!a   !b

?a $\tau@s$?b

(a@r)
(b@r)   B

$A = !a;A \oplus !b;A \oplus ?b;B$
$B = ?a;A \oplus \tau_{(s)};A$

$B \to^s A$
$A+B \to^r A+A$
$A+A \to^r B+B$

!a

?a $\tau@s$?b !b

(a@r)
(b@r/2)   B

$A = !a;A \oplus !b;B \oplus ?b;B$
$B = ?a;A \oplus \tau_{(s)};A$

Different reactions,
but they induce the
same ODEs

Luca Cardelli

Discrete-State Semantics

Discrete Semantics of Reactions

Syntax:

$$A+B \to^r A+A$$
$$A+B \to^r B+B$$
$$A+B+B$$

Semantics:

CTMC

Discrete Semantics of Reagents

Syntax:

Semantics:

$\{3B\} \quad \{1A,2B\} \; 2r_a \quad \quad 2r_a \quad \{3A\}$

$2r_b \quad 2r_b \quad \{2A,1B\}$

CTMC

ODE $=$ ODE

Continuous Chemistry

Process Algebra

Discrete Chemistry

CTMC $=$ CTMC
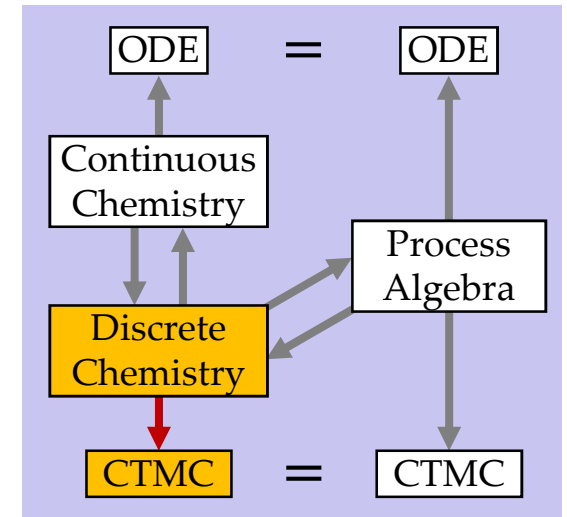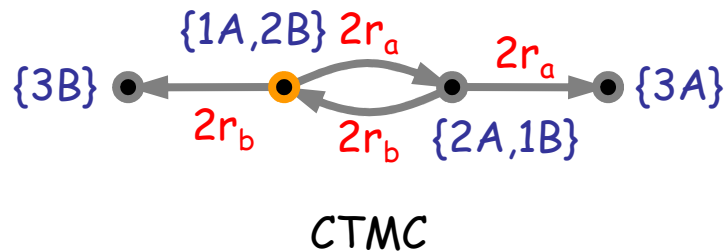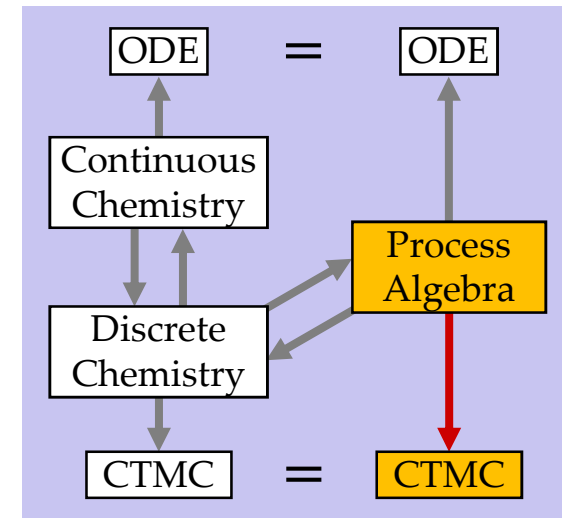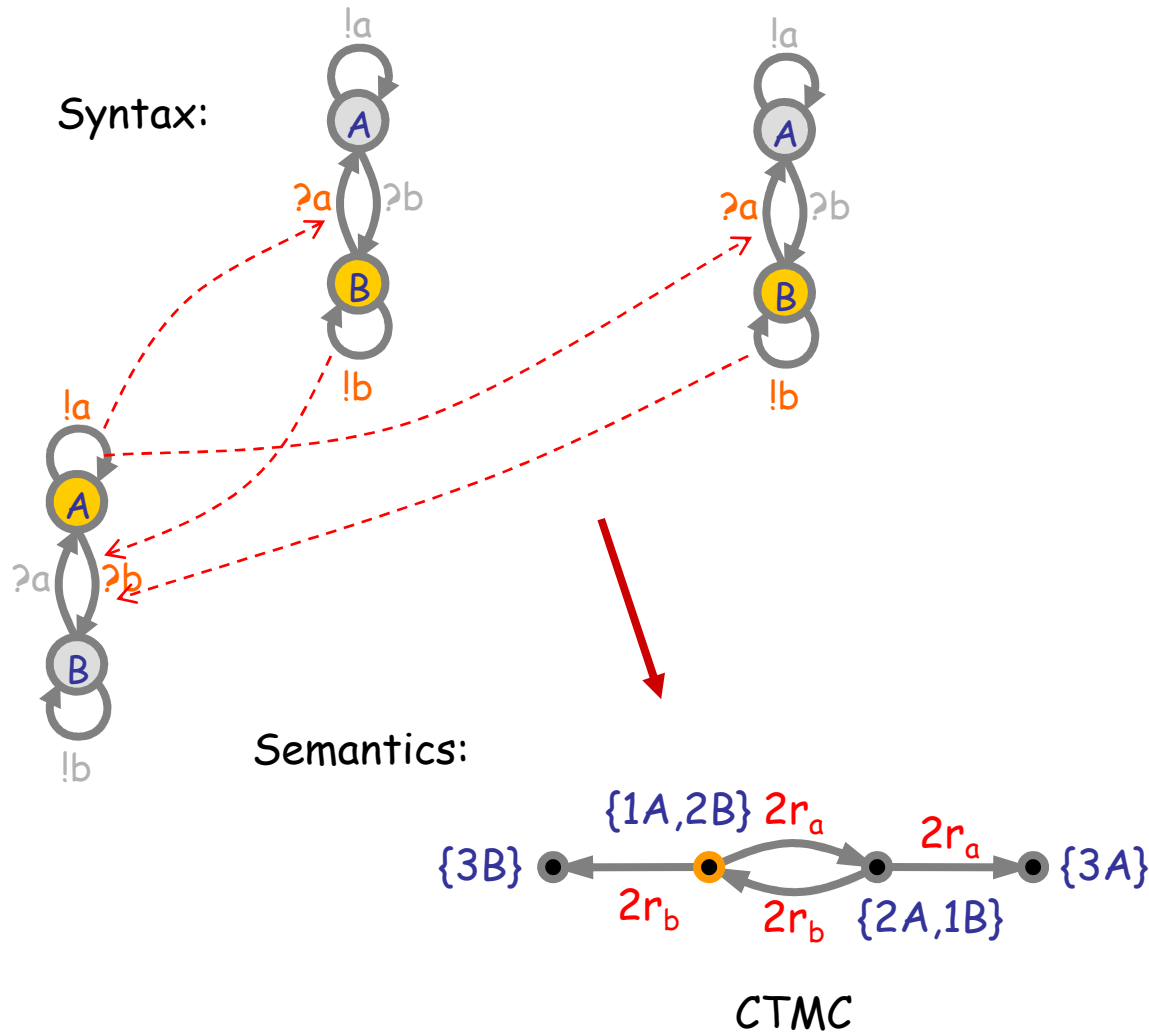
Discrete State Equivalence

- Def: ≋ is equivalent CTMC's (isomorphic graphs with same rates).

- Thm: E ≋ Ch(E)

- Thm: C ≋ Pi(C)
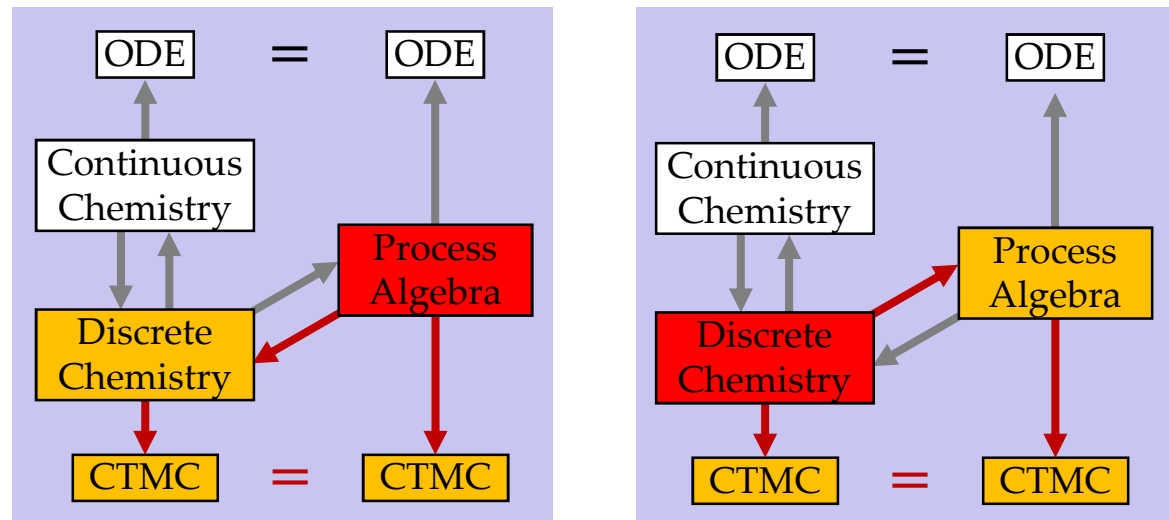
- For each E there is an E' ≋ E that is detangled (E' = Pi(Ch(E)))

- For each E in automata form there is an an E' ≋ E that is detangled and in automata form (E' = Detangle(E)).

Process Algebra = Discrete Chemistry

This is enough to establish that the process algebra is really faithful to the chemistry.

But CTMC are not the "ultimate semantics" because there are still questions of when two different CTMCs are actually equivalent (e.g. "lumping").

The "ultimate semantics" of chemistry is the *Chemical Master Equation* (derivable from the Chapman-Kolmogorov equation of the CTMC).
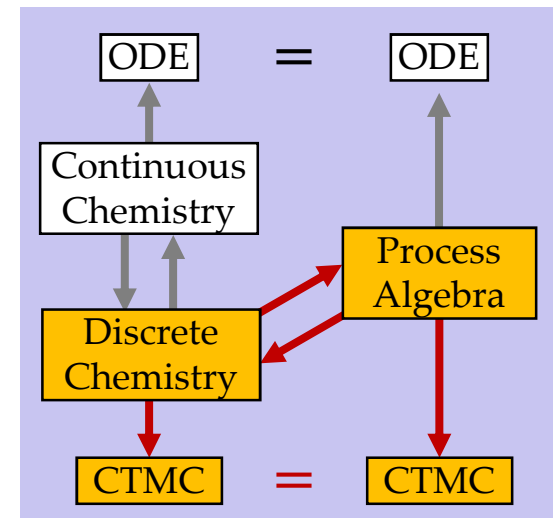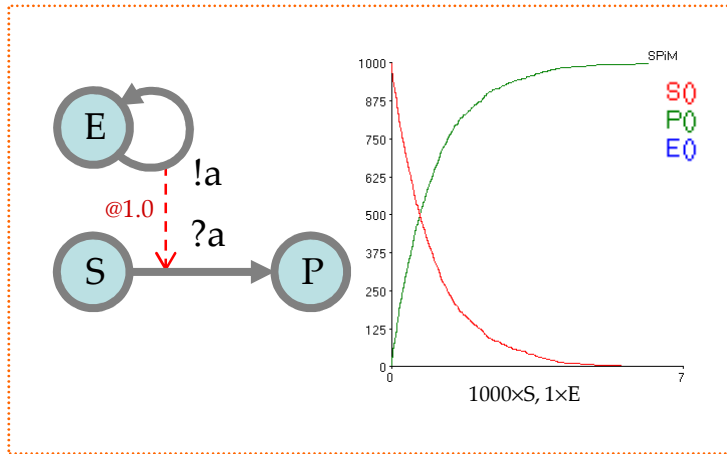
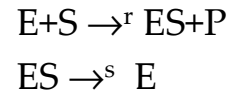# Exercise: Making Lines

Or: build me a population like this:
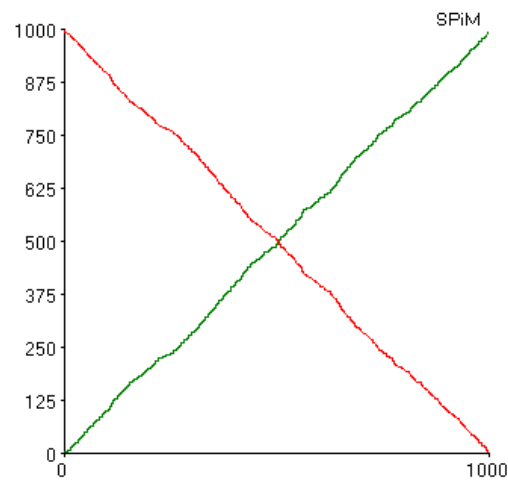
Second-order and Zero-order Regime

$E+S \to^r E+P$
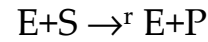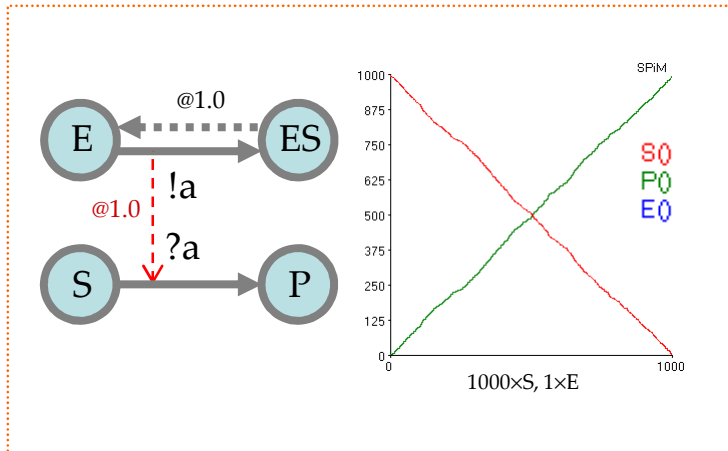
```
directive sample 1000.0
directive plot S(); P(); E()

new a@1.0:chan()

let E() = !a; E()
and S() = ?a; P()
and P() = ()

run (1 of E() | 1000 of S())
```
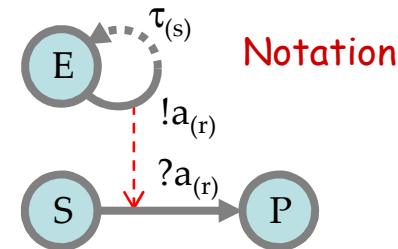
Second-Order Regime
$[S]^\bullet = -r[E][S]$

$E+S \to^r ES+P$
$ES \to^s E$

```
directive sample 1000.0
directive plot S(); P(); E()

new a@1.0:chan()

let E() = !a; delay@1.0; E()
and S() = ?a; P()
and P() = ()

run (1 of E() | 1000 of S())
```
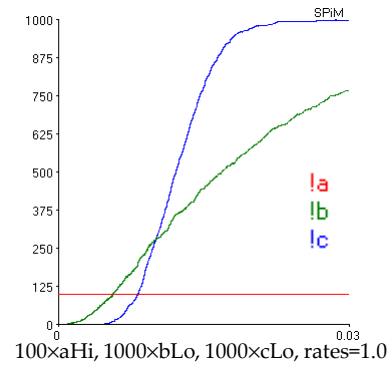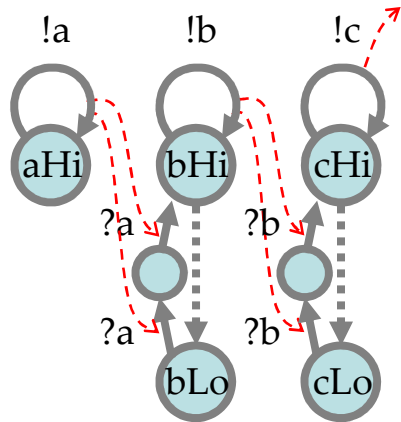
Zero-Order Regime
$[S]^\bullet \cong -1$ (by assuming $[ES]^\bullet = 0$)

Notation

2008-03-13

Cascades

!a   !b   !c

aHi   bHi   cHi

?a   ?b

?a   ?b

bLo   cLo

100×aHi, 1000×bLo, 1000×cLo, rates=1.0

Second-Oder Regime cascade:
a signal amplifier (MAPK)

$aHi > 0 \Rightarrow cHi = max$
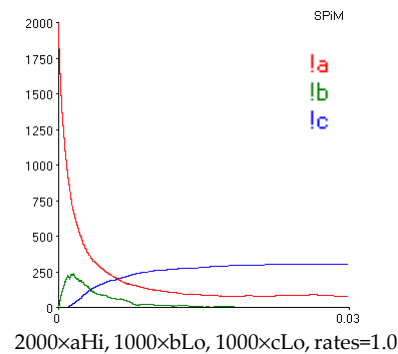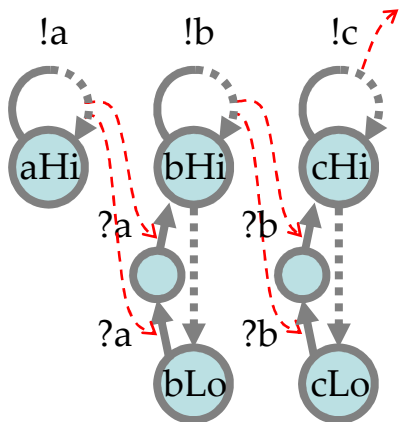
```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; A()
run 100 of A()
```

!a   !b   !c

aHi   bHi   cHi

?a   ?b

?a   ?b

bLo   cLo

2000×aHi, 1000×bLo, 1000×cLo, rates=1.0

Zero-Oder Regime cascade:
a signal *divider!*

$aHi = max \Rightarrow cHi = 1/3\ max$

```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; delay@1.0; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; delay@1.0; A()
run 2000 of A()
```
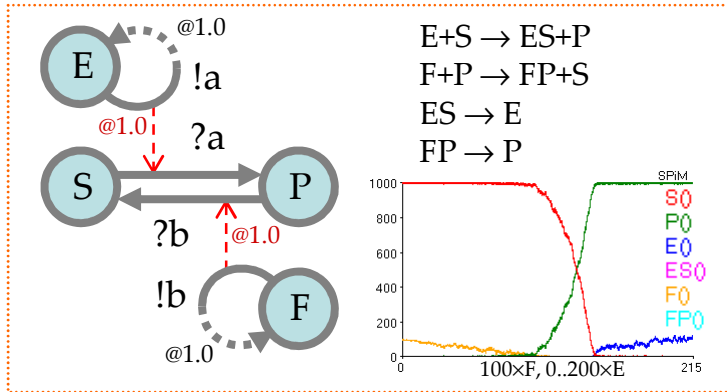
Ultrasensitivity

$$E+S \rightarrow ES+P$$
$$F+P \rightarrow FP+S$$
$$ES \rightarrow E$$
$$FP \rightarrow P$$

```
directive sample 215.0
directive plot S(); P(); E(); ES(); F(); FP()

new a@1.0:chan() new b@1.0:chan()

let S() = ?a; P()
and P() = ?b; S()

let E() = !a; delay@1.0; E()
and F() = !b; delay@1.0; F()

run 1000 of S()

let clock(t:float, tick:chan) =       (* sends a tick every t time *)
   (val ti = t/100.0 val d = 1.0/ti    (* by 100-step erlang timers *)
    let step(n:int) = if n<=0 then !tick; clock(t,tick) else delay@d; step(n-1)
    run step(100))
let Sig(p:proc(), tick:chan) = (p() | ?tick; Sig(p,tick))
let raising(p:proc(), t:float) =
   (new tick:chan run (clock(t,tick) | Sig(p,tick)))

run 100 of F()
run raising(E,1.0)
```
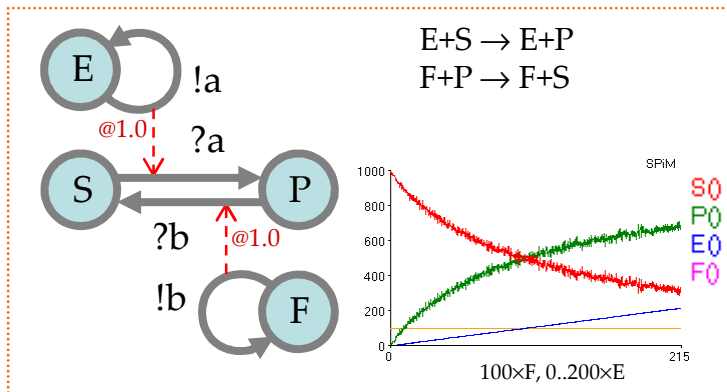
Zero-Order Regime
A small E-F inbalance causes
a much larger S-P switch.

$$E+S \rightarrow E+P$$
$$F+P \rightarrow F+S$$

```
directive sample 215.0 1000
directive plot S(); P(); E(); F()

new a@1.0:chan() new b@1.0:chan()

let S() = ?a; P()
and P() = ?b; S()

let E() = !a; E()
and F() = !b; F()

run 1000 of S()

let clock(t:float, tick:chan) =       (* sends a tick every t time *)
   (val ti = t/100.0 val d = 1.0/ti    (* by 100-step erlang timers *)
    let step(n:int) = if n<=0 then !tick; clock(t,tick) else delay@d; step(n-1)
    run step(100))
let Sig(p:proc(), tick:chan) = (p() | ?tick; Sig(p,tick))
let raising(p:proc(), t:float) =
   (new tick:chan run (clock(t,tick) | Sig(p,tick)))

run 100 of F()
run raising(E,1.0)
```

Second-Order Regime

2008-03-13   41

Continuous-State Semantics
(short version)
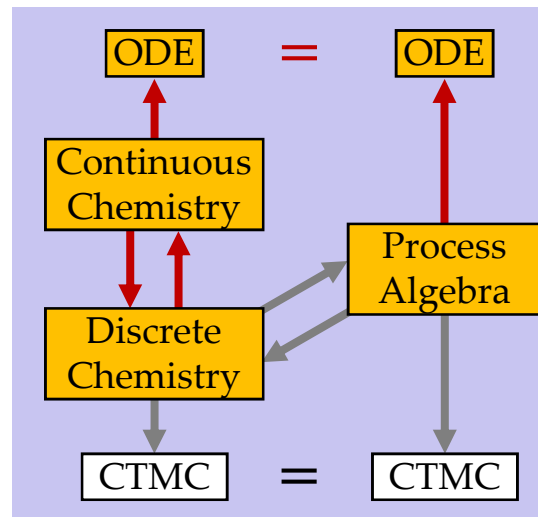
The Gillespie$^{(?)}$ Conversion

Discrete Chemistry	Continuous Chemistry	$\gamma = N_A V$	:M^{-1}
initial quantities $\#A_0$	initial concentrations $[A]_0$	with $[A]_0 = \#A_0/\gamma$	
$A \dashrightarrow^r A'$	$A \to^k A'$	with $k = r$	:s^{-1}
$A+B \dashrightarrow^r A'+B'$	$A+B \to^k A'+B'$	with $k = r\gamma$	:M^{-1}s^{-1}
$A+A \dashrightarrow^r A'+A''$	$A+A \to^k A'+A''$	with $k = r\gamma/2$	:M^{-1}s^{-1}

V = interaction volume
N_A = Avogadro's number

Think $\gamma = 1$
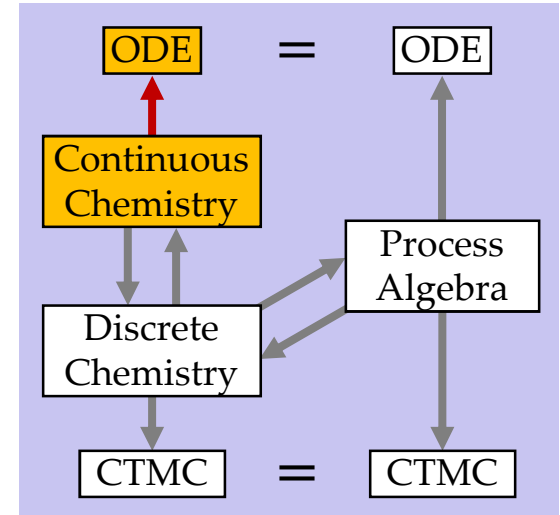i.e. $V = 1/N_A$

M = *mol·L^{-1}*
    molarity (concentration)

ODE $=$ ODE

Continuous Chemistry

Process Algebra

Discrete Chemistry

CTMC $=$ CTMC
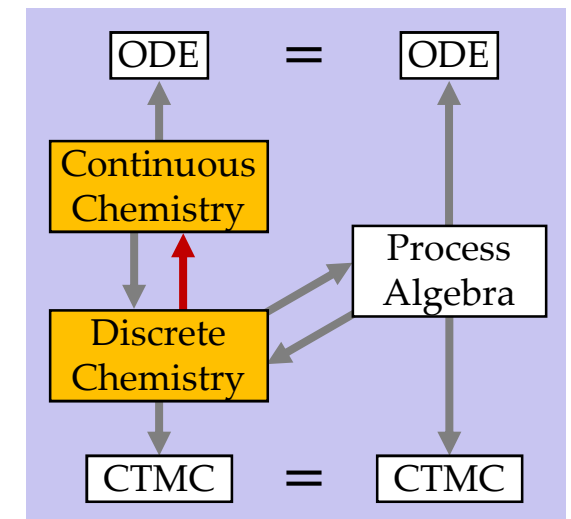
From Reactions to ODEs

v_1: $A+B \rightarrow_{k_1} C+C$

v_2: $A+C \rightarrow_{k_2} D$

v_3: $C \rightarrow_{k_3} E+F$

v_4: $F+F \rightarrow_{k_4} B$
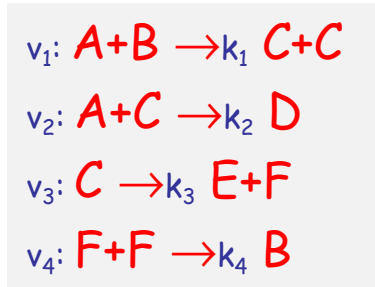
Write the coefficients by columns

Stoichiometric Matrix
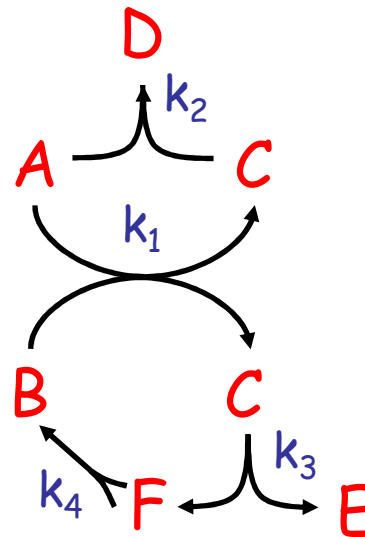
Quantity changes

Stoichiometric matrix

Rate laws

$$d[\mathbf{X}]/dt = \mathbf{N} \cdot \mathbf{l}$$

reactions

N	v_1	v_2	v_3	v_4
A	-1	-1		
B	-1			1
C	2	-1	-1	
D		1		
E			1	
F			1	-2

species

X

| ODE | = | ODE |

| Continuous Chemistry |

| Discrete Chemistry |

| Process Algebra |

| CTMC | = | CTMC |

$$d[\mathbf{A}]/dt = -l_1 - l_2$$
$$d[\mathbf{B}]/dt = -l_1 + l_4$$
$$d[\mathbf{C}]/dt = 2l_1 - l_2 - l_3$$
$$d[\mathbf{D}]/dt = l_2$$
$$d[\mathbf{E}]/dt = l_3$$
$$d[\mathbf{F}]/dt = l_3 - 2l_4$$

Read the concentration changes from the rows

E.g. $d[\mathbf{A}]/dt = -k_1[\mathbf{A}][\mathbf{B}] - k_2[\mathbf{A}][\mathbf{C}]$

Set a rate law for each reaction (Degradation/Hetero/Homeo)

l	
l_1	$k_1[\mathbf{A}][\mathbf{B}]$
l_2	$k_2[\mathbf{A}][\mathbf{C}]$
l_3	$k_3[\mathbf{C}]$
l_4	$k_4[\mathbf{F}]^2$

X: chemical species
[-]: quantity of molecules
l: rate laws
k: kinetic parameters
N: stoichiometric matrix

Processes Rate Equation

Process Rate Equation for Reagents E in volume γ

$$d[X]/dt = (\Sigma(Y \in E)\ Accr_E(Y,X)\cdot[Y]) - Depl_E(X)\cdot[X]$$
$$\text{for all } X \in E$$

"The change in process concentration (!!) for X at time t is:
   the sum over all possible (kinds of) processes Y of:
      the concentration at time t of Y
         times the accretion from Y to X
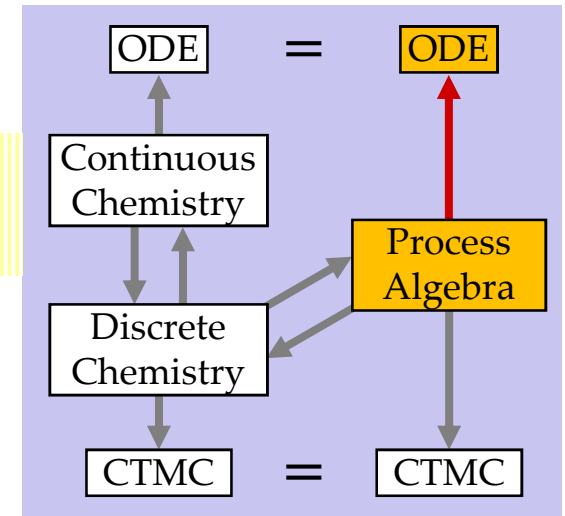   minus the concentration at time t of X
      times the depletion of X to some other Y"

$Depl_E(X) =$
   $\Sigma(i: E.X.i=\tau_{(r)};P)\ r\ +$
   $\Sigma(i: E.X.i=?a_{(r)};P)\ r\gamma\cdot OutsOn_E(a)\ +$
   $\Sigma(i: E.X.i=!a_{(r)};P)\ r\gamma\cdot InsOn_E(a)$

$Accr_E(Y, X) =$
   $\Sigma(i: E.Y.i=\tau_{(r)};P)\ \#X(P)\cdot r\ +$
   $\Sigma(i: E.Y.i=?a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot OutsOn_E(a)\ +$
   $\Sigma(i: E.Y.i=!a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot InsOn_E(a)$

$InsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=?a_{(r)};P\}\cdot[Y]$
$OutsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=!a_{(r)};P\}\cdot[Y]$

ODE	=	ODE

Continuous Chemistry — Process Algebra

Discrete Chemistry

CTMC	=	CTMC

$X = \tau_{(r)};0 \longrightarrow d[X]/dt = -r[X]$

$X = ?a_{(r)};0 \quad\quad d[X]/dt = -r\gamma[X][Y]$
$Y = !a_{(r)};0 \longrightarrow d[Y]/dt = -r\gamma[X][Y]$

$X = ?a_{(r)};0 \longrightarrow d[X]/dt = -2r\gamma[X]^2$
$\quad\oplus !a_{(r)};0$

Continuous State Equivalence

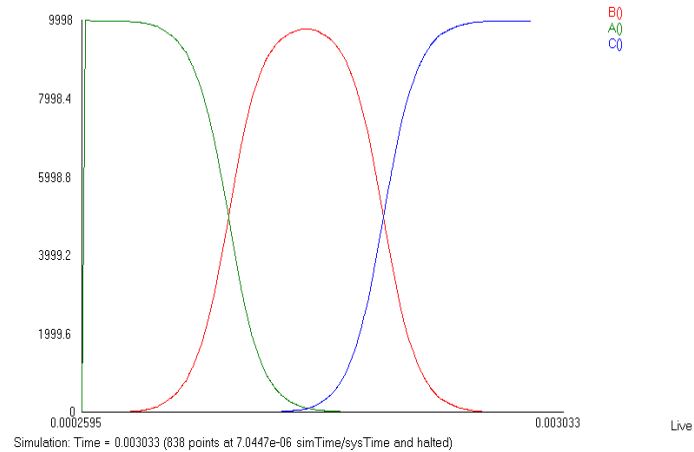- Def: ≈ is equivalence of polynomials over the field of reals.
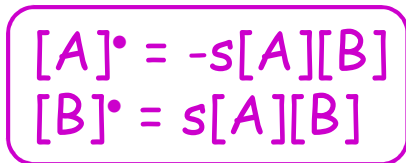
- Thm: E ≈ Cont(Ch(E))

- Thm: Cont(C) ≈ Pi(C)
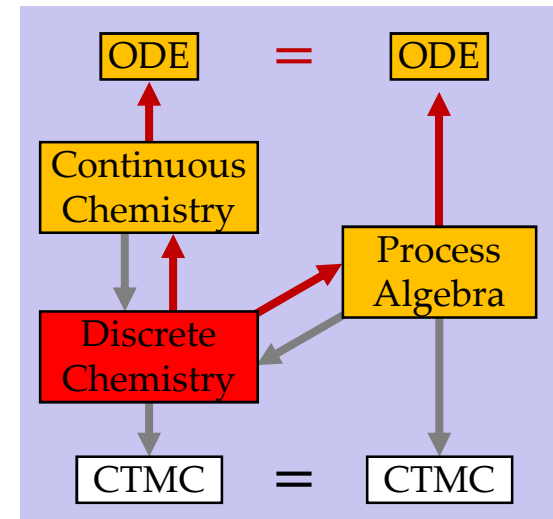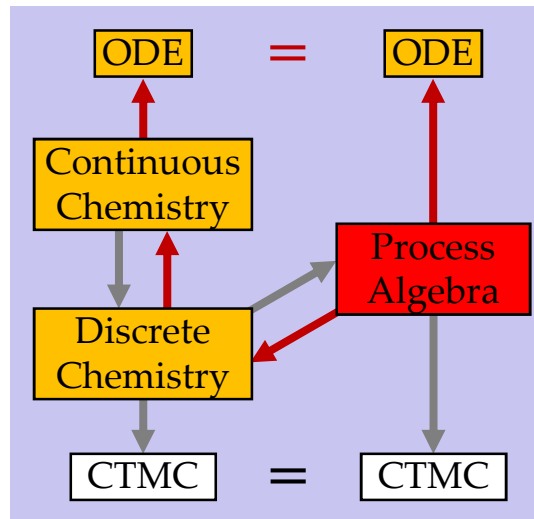
- For each E there is an E' ≈ E that is detangled (E' = Pi(Ch(E)))

- For each E in automata form there is an an E' ≈ E that is detangled and in automata form (E' = Detangle(E)).

# Exercise: Making Waves

Or: build me a population like this:

BO
AO
CO

Simulation: Time = 0.003033 (838 points at 7.0447e-06 simTime/sysTime and halted)

Live

Nonlinear Transition (NLT)

!c

@s

?c

(A) → (B)

$A = ?c_{(s)};B$
$B = !c_{(s)};B$

↓

$A+B \rightarrow^s B+B$

↓

$[A]^\bullet = -s[A][B]$
$[B]^\bullet = s[A][B]$

999

666

333

0
0.0015957   0.014324   Paused

SPiM

Simulation: Time = 0.013448 (999 points at 0.0085215 simTime/sysTime and halted)

B()
A()
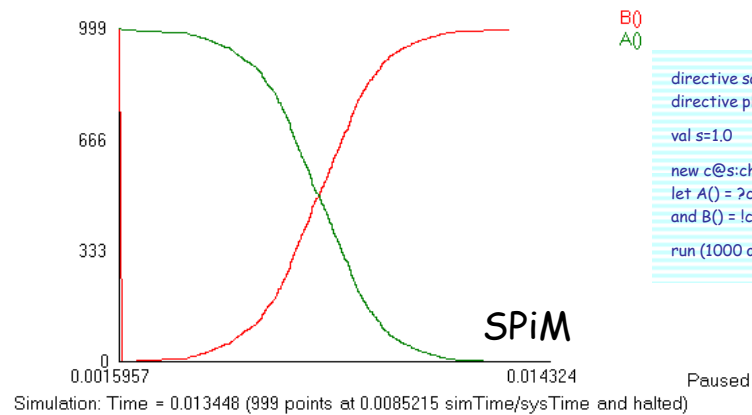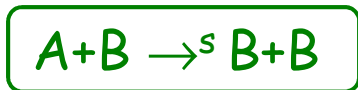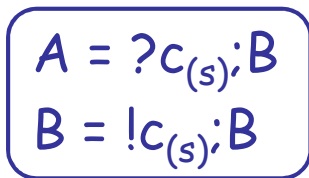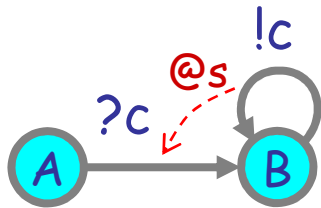
directive sample 0.02 1000
directive plot B(); A()

val s=1.0

new c@s:chan
let A() = ?c; B()
and B() = !c;B()

run (1000 of A() | 1 of B())

N.B.: needs at
least 1 B to
"get started".

Matlab
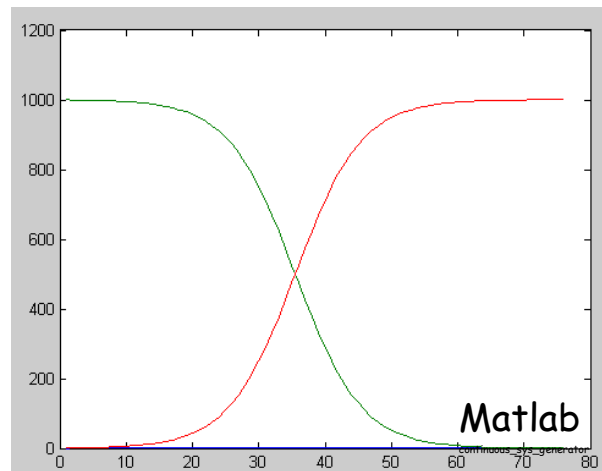
interval/step [0:0.001:0.0]
(A)   dx1/dt = - x1*x2   1000.0
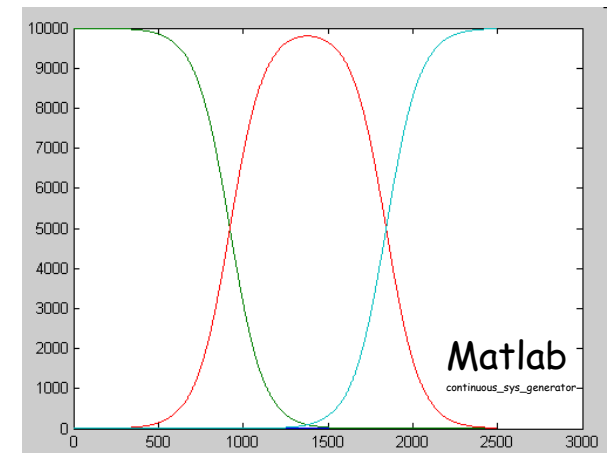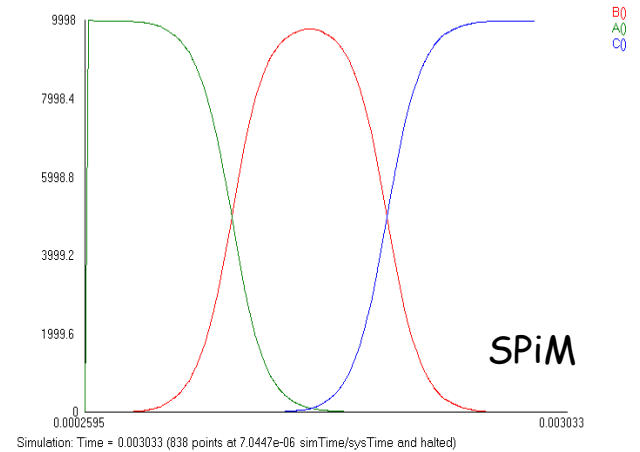(B)   dx2/dt = x1*x2   1.0

Two NLTs: Bell Shape

!b   !c

?b   ?c

A → B → C

$[B]^\bullet = [B]([A]-[C])$
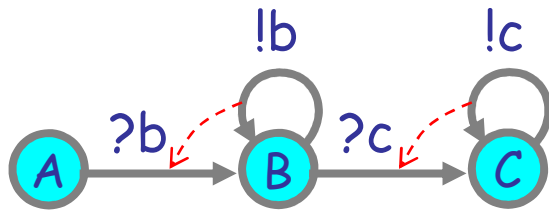
$A = ?b_{(1)};B$
$B = !b_{(1)};B \oplus ?c_{(1)};C$
$C = !c_{(1)};C$

$A+B \rightarrow^1 B+B$
$B+C \rightarrow^1 C+C$

$[A]^\bullet = -[A][B]$
$[B]^\bullet = [A][B]-[B][C]$
$[C]^\bullet = [B][C]$
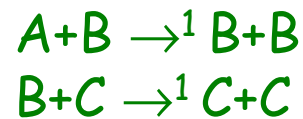
directive sample 0.0025 1000
directive plot B(); A(); C()

new b@1.0:chan new c@1.0:chan

let A() = ?b; B()
and B() = do !b;B() or ?c; C()
and C() = !c;C()

run ((10000 of A()) | B() | C())

9998
7998.4
5998.8
3999.2
1999.6
0
0.0002595   0.003033

B()
A()
C()

SPiM

Simulation: Time = 0.003033 (838 points at 7.0447e-06 simTime/sysTime and halted)

10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0   500   1000   1500   2000   2500   3000

Matlab
continuous_sys_generator

interval/step [0:0.000001:0.0025]
(A)   dx1/dt = -x1*x2   10000.0
(B)   dx2/dt = x1*x2 - x2*x3   1.0
(C)   dx3/dt = x2*x3   1.0

NLT in a Cycle: Oscillator (unstable)

!c

C

?a   @1.0

@1.0 →   ?c

A → B

!a   ?b   !b
      @1.0

SPiM

A()
B()
C()

900xA, 500xB, 100xC
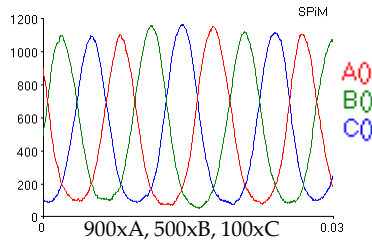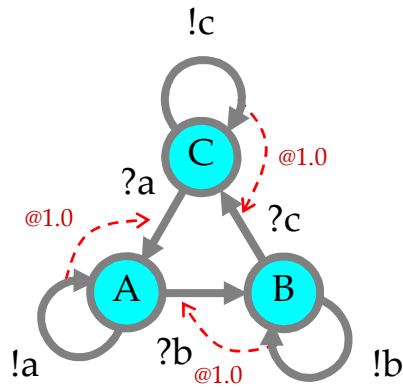
directive sample 0.03 1000
directive plot A(); B(); C()
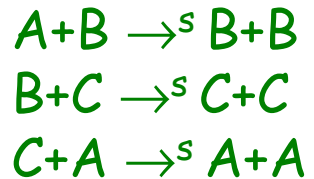
new a@1.0:chan new b@1.0:chan new c@1.0:chan
let A() = do !a;A() or ?b; B()
and B() = do !b;B() or ?c; C()
and C() = do !c;C() or ?a; A()

run (900 of A() | 500 of B() | 100 of C())

$A = !a_{(s)};A \oplus ?b_{(s)};B$
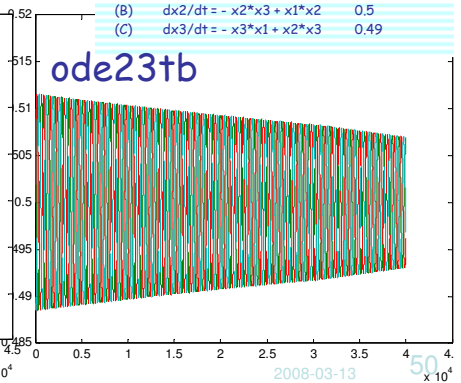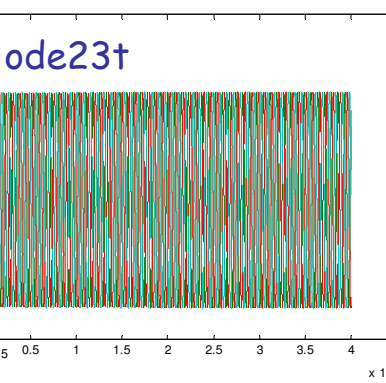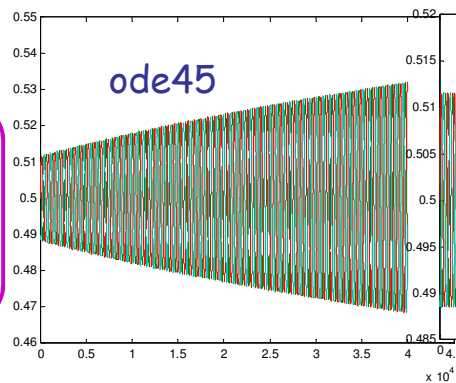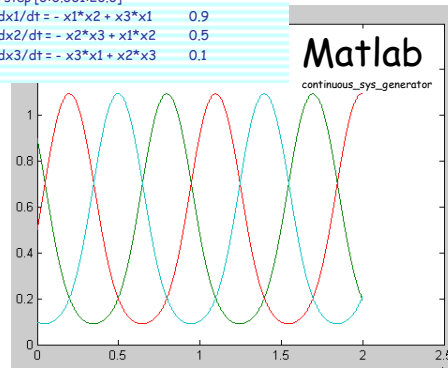$B = !b_{(s)};B \oplus ?c_{(s)};C$
$C = !c_{(s)};C \oplus ?a_{(s)};A$

$A+B \rightarrow^s B+B$
$B+C \rightarrow^s C+C$
$C+A \rightarrow^s A+A$

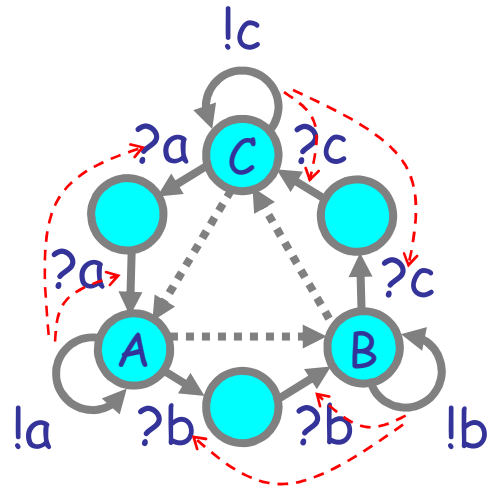$[A]^\bullet = -s[A][B]+s[C][A]$
$[B]^\bullet = -s[B][C]+s[A][B]$
$[C]^\bullet = -s[C][A]+s[B][C]$

interval/step [0:0.001:20.0]
(A)   dx1/dt = - x1*x2 + x3*x1   0.9
(B)   dx2/dt = - x2*x3 + x1*x2   0.5
(C)   dx3/dt = - x3*x1 + x2*x3   0.1

Matlab
continuous_sys_generator

interval/step [0:0.01:400.0]
(A)   dx1/dt = - x1*x2 + x3*x1   0.51
(B)   dx2/dt = - x2*x3 + x1*x2   0.5
(C)   dx3/dt = - x3*x1 + x2*x3   0.49

ode45

ode23t

ode23tb

Oscillator (stable)


```
directive sample 0.1 1000
directive plot A1(); A2(); A3()

val r=1.0 val s=1.0

new a1@s:chan new a2@s:chan new a3@s:chan
let A1() = do !a1;A1() or delay@r;A2() or ?a2; ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3; ?a3; A3()
and A3() = do !a3;A3() or delay@r;A1() or ?a1; ?a1; A1()

run 1000 of A1()
```
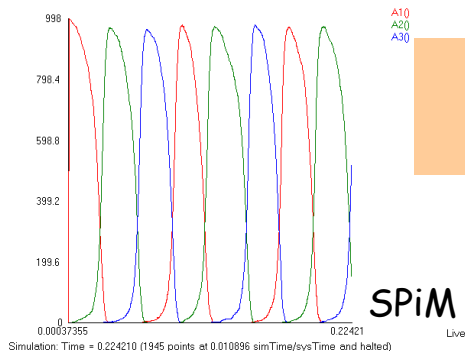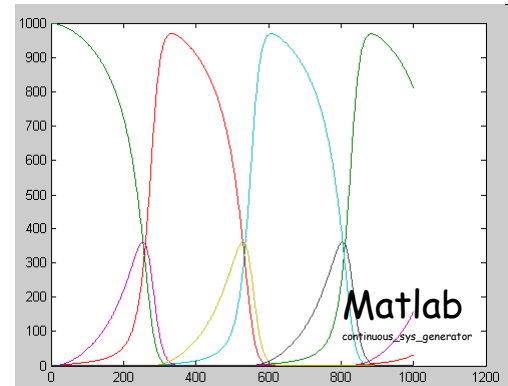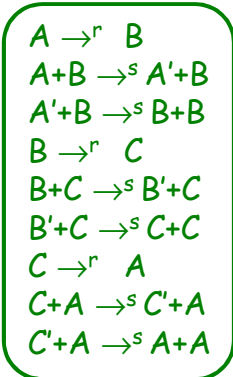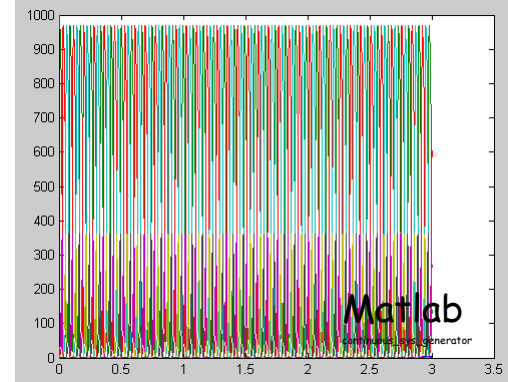
N.B. this does not deadlock!

$A = !a_{(s)};A \oplus \tau_r;B \oplus ?b_{(s)};A'$
$A' = ?b_{(s)};B$
$B = !b_{(s)};B \oplus \tau_r;C \oplus ?c_{(s)};B'$
$B' = ?c_{(s)};C$
$C = !c_{(s)};C \oplus \tau_r;A \oplus ?a_{(s)};C'$
$C' = ?a_{(s)};A$

Sustained Deterministic Oscillation

$A \rightarrow^r B$
$A+B \rightarrow^s A'+B$
$A'+B \rightarrow^s B+B$
$B \rightarrow^r C$
$B+C \rightarrow^s B'+C$
$B'+C \rightarrow^s C+C$
$C \rightarrow^r A$
$C+A \rightarrow^s C'+A$
$C'+A \rightarrow^s A+A$

Matlab
continuous_sys_generator

Robust Stochastic Oscillation

SPiM
Live
Simulation: Time = 0.224210 (1945 points at 0.010896 simTime/sysTime and halted)

$[A]^{\bullet} = -r[A]-s[A][B]+r[C]+s[C'][A]$
$[B]^{\bullet} = -r[B]-s[B][C]+r[A]+s[A'][B]$
$[C]^{\bullet} = -r[C]-s[C][A]+r[B]+s[B'][C]$
$[A']^{\bullet} = -s[A'][B] + s[A][B]$
$[B']^{\bullet} = -s[B'][C] + s[B][C]$
$[C']^{\bullet} = -s[C'][A] + s[C][A]$

Matlab

SPiM
Live
Simulation: Time = 0.729952 (6432 points at 0.010541 simTime/sysTime and halted)

```
interval/step [0:0.0001:0.1]
(A)    dx1/dt = -x1 - x1*x2 + x3 + x6*x1     1000.0
(B)    dx2/dt = -x2 - x2*x3 + x1 + x4*x2     0.0
(C)    dx3/dt = -x3 - x3*x1 + x2 + x5*x3     0.0
(A')   dx4/dt = -x4*x2 + x1*x2               0.0
(B')   dx5/dt = -x5*x3 + x2*x3               0.0
(C')   dx6/dt = -x6*x1 + x3*x1               0.0
```

NLTs in Series: Soliton Propagation
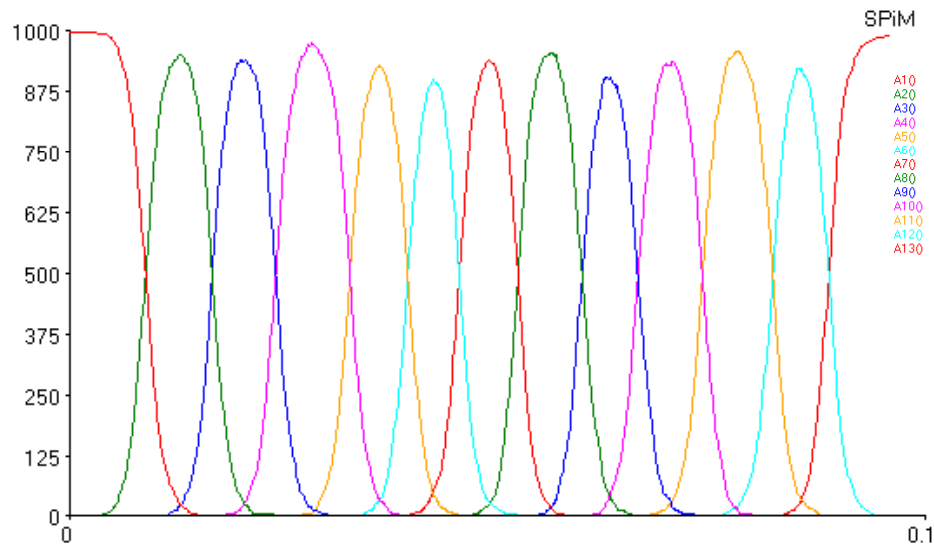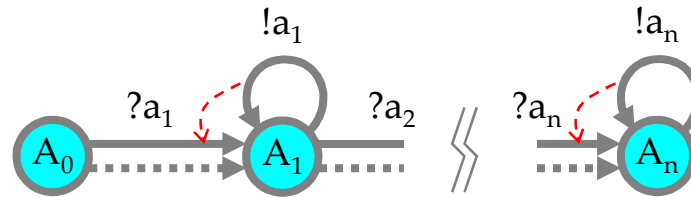

```
directive sample 0.1 1000
directive plot A1(); A2(); A3(); A4(); A5(); A6(); A7(); A8();
A9(); A10(); A11(); A12(); A13()

val r=1.0 val s=1.0

new a2@s:chan new a3@s:chan new a4@s:chan
new a5@s:chan new a6@s:chan new a7@s:chan
new a8@s:chan new a9@s:chan new a10@s:chan
new a11@s:chan new a12@s:chan new a13@s:chan
let A1() = do delay@r;A2() or ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3; A3()
and A3() = do !a3;A3() or delay@r;A4() or ?a4; A4()
and A4() = do !a4;A4() or delay@r;A5() or ?a5; A5()
and A5() = do !a5;A5() or delay@r;A6() or ?a6; A6()
and A6() = do !a6;A6() or delay@r;A7() or ?a7; A7()
and A7() = do !a7;A7() or delay@r;A8() or ?a8; A8()
and A8() = do !a8;A8() or delay@r;A9() or ?a9; A9()
and A9() = do !a9;A9() or delay@r;A10() or ?a10; A10()
and A10() = do !a10;A10() or delay@r;A11() or ?a11; A11()
and A11() = do !a11;A11() or delay@r;A12() or ?a12; A12()
and A12() = do !a12;A12() or delay@r;A13() or ?a13; A13()
and A13() = !a13;A13()

run 1000 of A1()
```
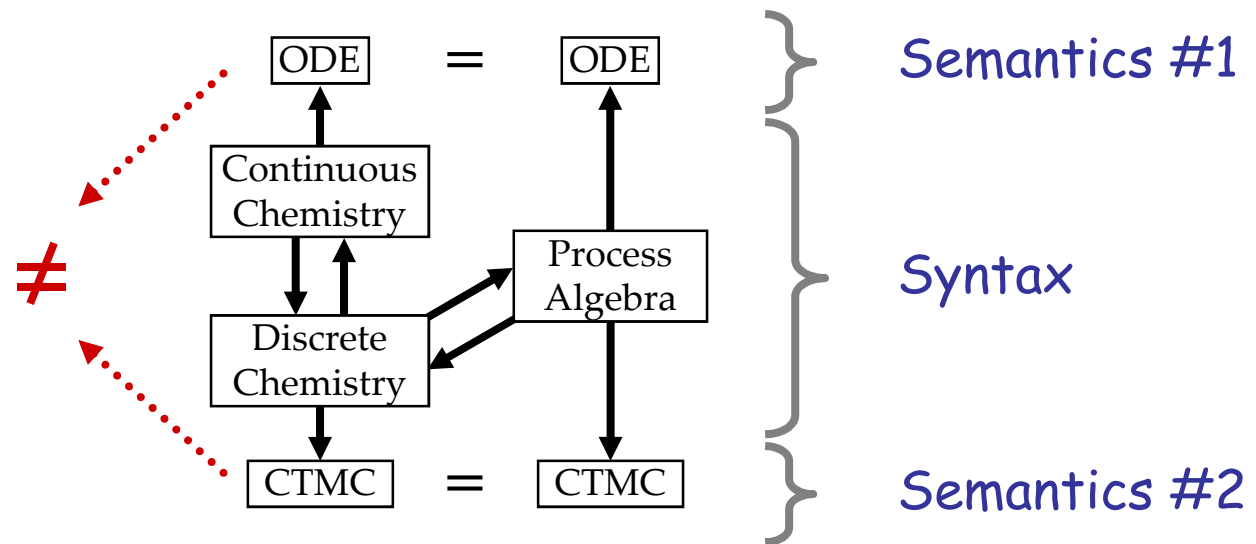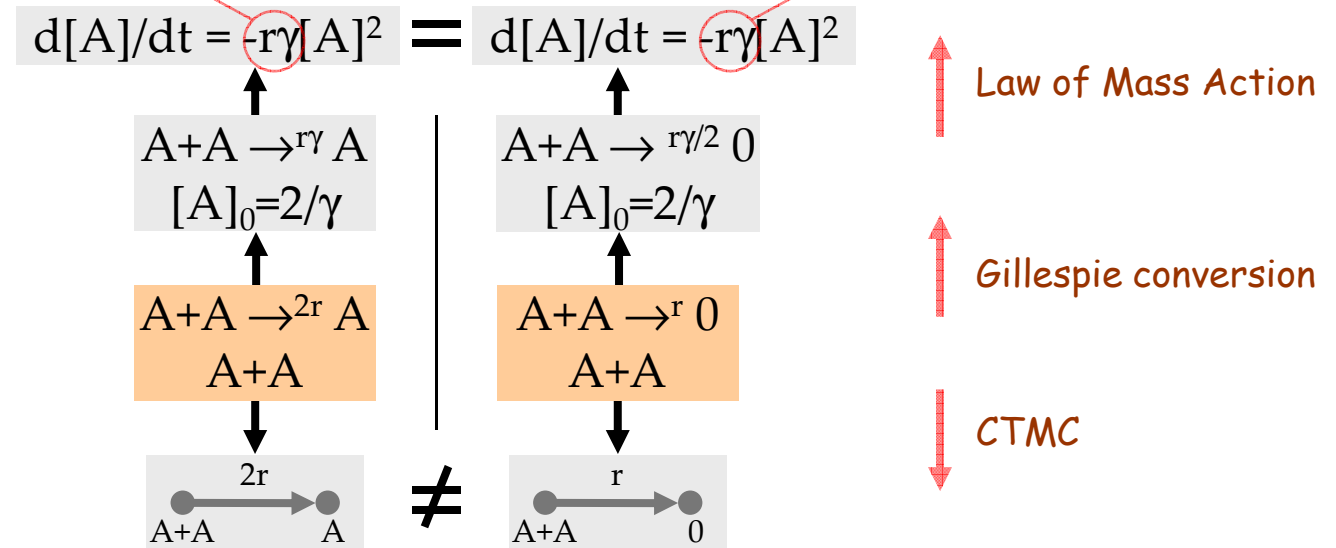
GMA ≠ CME

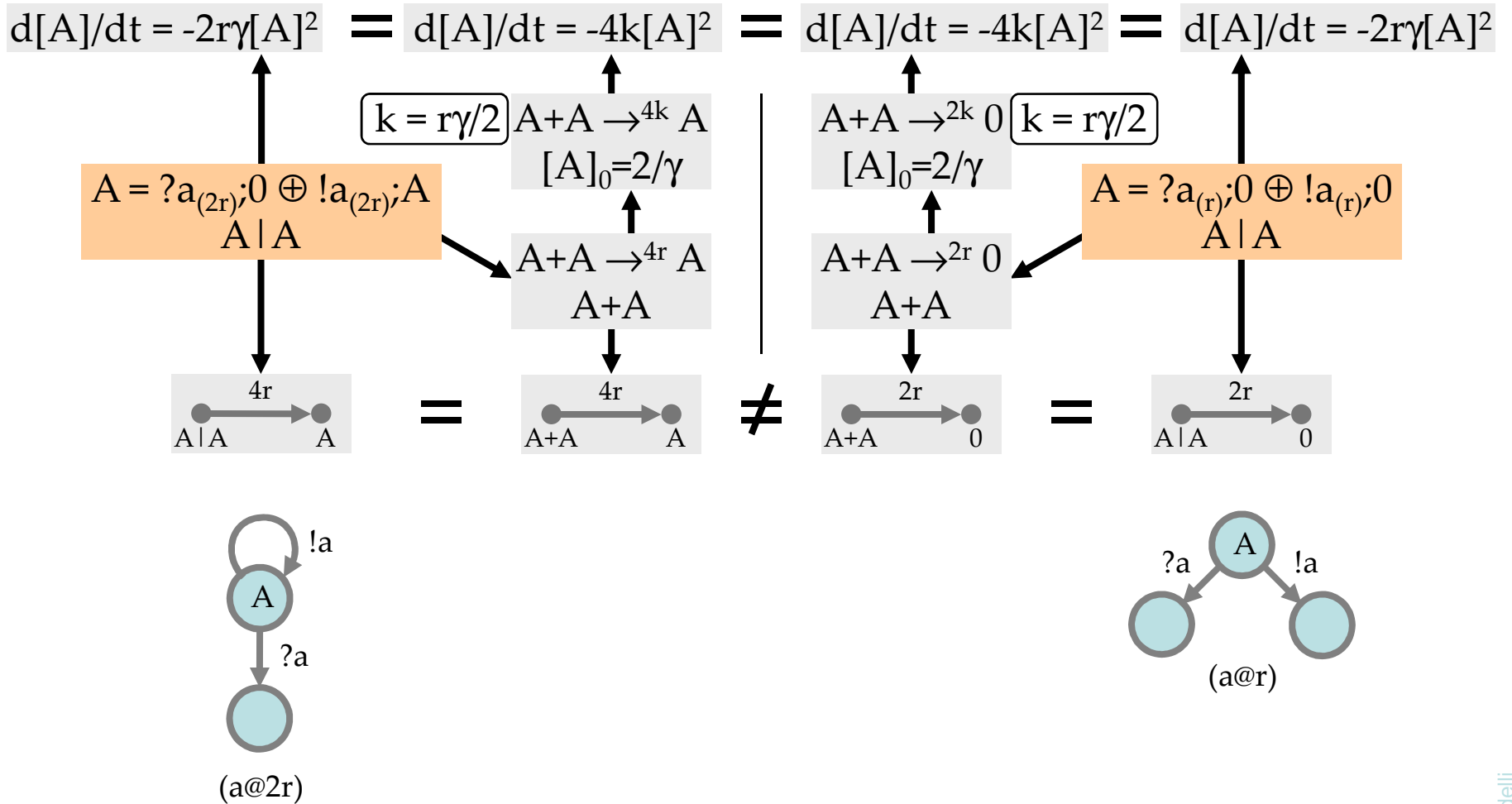$$\mathbf{A+A \to^{2r} A} \quad \color{red}{=?} \quad \mathbf{A+A \to^{r} 0}$$
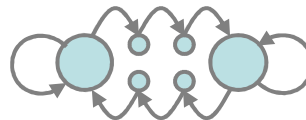
1*reaction rate rγ because
1*A is lost in reaction.

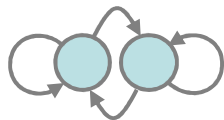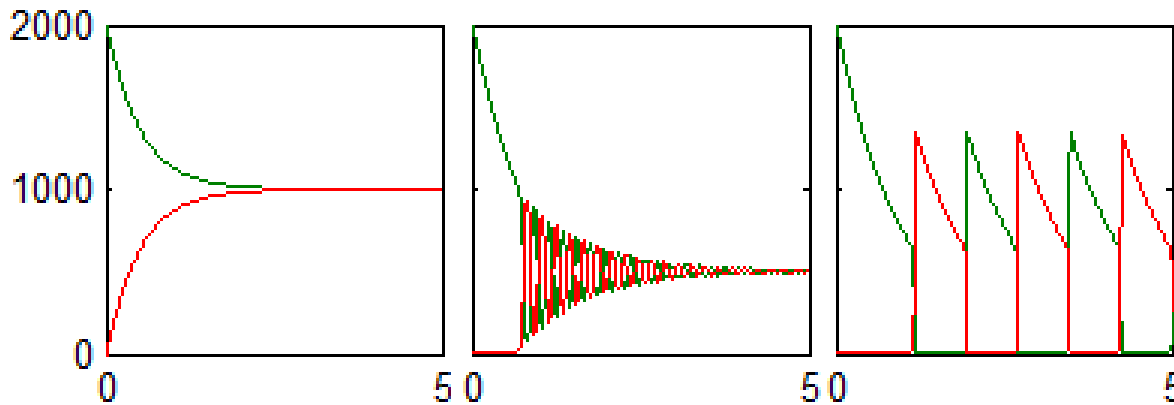2*reaction rate rγ/2 because
2*A are lost in reaction.

$$d[A]/dt = -r\gamma[A]^2 \quad = \quad d[A]/dt = -r\gamma[A]^2$$

Law of Mass Action

$$A+A \to^{r\gamma} A$$
$$[A]_0 = 2/\gamma$$

$$A+A \to^{r\gamma/2} 0$$
$$[A]_0 = 2/\gamma$$

Gillespie conversion

$$A+A \to^{2r} A$$
$$A+A$$

$$A+A \to^{r} 0$$
$$A+A$$

CTMC

2r
A+A ●——→● A

$$\ne$$

r
A+A ●——→● 0

... as Automata

$$d[A]/dt = -2r\gamma[A]^2 \quad = \quad d[A]/dt = -4k[A]^2 \quad = \quad d[A]/dt = -4k[A]^2 \quad = \quad d[A]/dt = -2r\gamma[A]^2$$

$k = r\gamma/2$

$A+A \to^{4k} A$
$[A]_0 = 2/\gamma$

$A = ?a_{(2r)};0 \oplus !a_{(2r)};A$
$A | A$

$A+A \to^{4r} A$
$A+A$

$A+A \to^{2k} 0$
$[A]_0 = 2/\gamma$

$k = r\gamma/2$

$A+A \to^{2r} 0$
$A+A$

$A = ?a_{(r)};0 \oplus !a_{(r)};0$
$A | A$

4r
A|A → A

=

4r
A+A → A

≠

2r
A+A → 0

=

2r
A|A → 0

!a
A
?a
(a@2r)

?a A !a
(a@r)

Continuous vs. Discrete Groupies
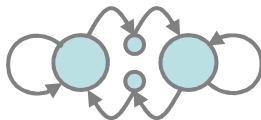
(all with doping)

Matlab

SPiM

B()
A()

$2000 \times A , 0 \times B , 1 \times A_d , 1 \times B_d , r = 1.0$

And Yet It Moves

The Repressilator

x   **Neg**   z

y

Neg   **Neg**

Neg(a,b) $\tau_{(\varepsilon)}$ Tr(b)   !b

?a $\tau_{(\eta)}$ $\tau_{(\delta)}$

Inh(a,b)

A fine stochastic oscillator over a wide range of parameters.

123

61.5

SPiM

0

53.947   40054

Simulation: Time = 53810.179900 (1070 points at 34439 simTime/sysTime and halted)

!a
!b
!c

Paused
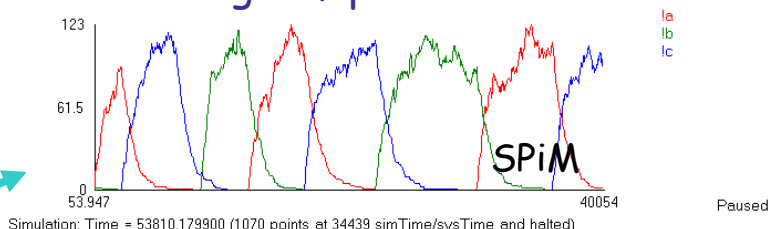
Parametric representation

Neg(a,b) = ?a; Inh(a,b) $\oplus$ τ_ε; (Tr(b) | Neg(a,b))

Inh(a,b) = τ_η; Neg(a,b)

Tr(b) = !b; Tr(b) $\oplus$ τ_γ; 0

Neg($x_{(r)}$,$y_{(r)}$) | Neg($y_{(r)}$,$z_{(r)}$) | Neg($z_{(r)}$,$x_{(r)}$)

Neg/x,y $\to^\varepsilon$ Tr/y + Neg/x,y
Neg/y,z $\to^\varepsilon$ Tr/z + Neg/y,z
Neg/z,x $\to^\varepsilon$ Tr/x + Neg/z,x
Tr/x + Neg/x,y $\to^r$ Tr/x + Inh/x,y
Tr/y + Neg/y,z $\to^r$ Tr/y + Inh/y,z
Tr/z + Neg/z,x $\to^r$ Tr/z + Inh/z,x
Inh/x,y $\to^\eta$ Neg/x,y
Inh/y,z $\to^\eta$ Neg/y,z
Inh/z,x $\to^\eta$ Neg/z,x
Tr/x $\to^\gamma$ 0
Tr/y $\to^\gamma$ 0
Tr/z $\to^\gamma$ 0
Neg/x,y + Neg/y,z + Neg/z,x

---

d[Neg/x,y]/dt = -r[Tr/x][Neg/x,y] + η[Inh/x,y]
d[Neg/y,z]/dt = -r[Tr/y][Neg/y,z] + η[Inh/y,z]
d[Neg/z,x]/dt = -r[Tr/z][Neg/z,x] + η[Inh/z,x]
d[Inh/x,y]/dt = r[Tr/x][Neg/x,y] - η[Inh/x,y]
d[Inh/y,z]/dt = r[Tr/y][Neg/y,z] - η[Inh/y,z]
d[Inh/z,x]/dt = r[Tr/z][Neg/z,x] - η[Inh/z,x]
d[Tr/x]/dt = ε[Neg/z,x] - γ[Tr/x]
d[Tr/y]/dt = ε[Neg/x,y] - γ[Tr/y]
d[Tr/z]/dt = ε[Neg/y,z] - γ[Tr/z]

simplifying (N is the quantity of each of the 3 gates)

d[Neg/x,y]/dt = ηN – (η+r[Tr/x])[Neg/x,y]
d[Neg/y,z]/dt = ηN – (η+r[Tr/y])[Neg/y,z]
d[Neg/z,x]/dt = ηN – (η+r[Tr/z])[Neg/z,x]
d[Tr/x]/dt = ε[Neg/z,x] - γ[Tr/x]
d[Tr/y]/dt = ε[Neg/x,y] - γ[Tr/y]
d[Tr/z]/dt = ε[Neg/y,z] - γ[Tr/z]

Analytically not an oscillator!

Matlab

continuous_sys_generator

100
90
80
70
60
50
40
30
20
10
0

0   500   1000   1500   2000   2500
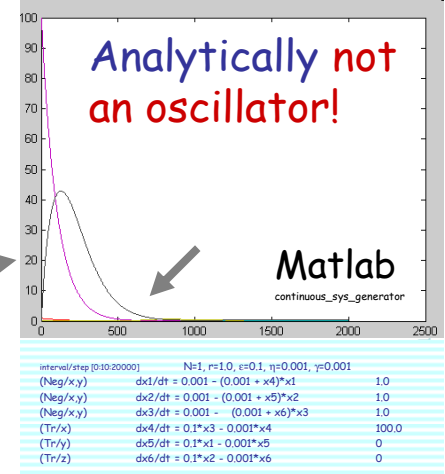
interval/step [0:10:20000]   N=1, r=1.0, ε=0.1, η=0.001, γ=0.001
(Neg/x,y)   dx1/dt = 0.001 - (0.001 + x4)*x1   1.0
(Neg/x,y)   dx2/dt = 0.001 - (0.001 + x5)*x2   1.0
(Neg/x,y)   dx3/dt = 0.001 -   (0.001 + x6)*x3   1.0
(Tr/x)   dx4/dt = 0.1*x3 - 0.001*x4   100.0
(Tr/y)   dx5/dt = 0.1*x1 - 0.001*x5   0
(Tr/z)   dx6/dt = 0.1*x2 - 0.001*x6   0

Model Compactness

Entangled vs detangled

E_3
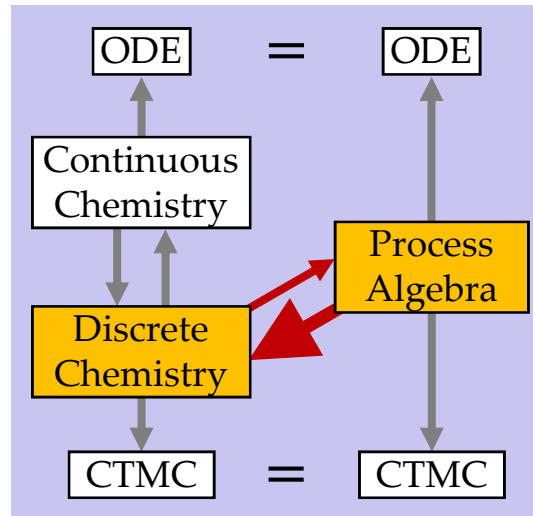
Detangle(E_3)
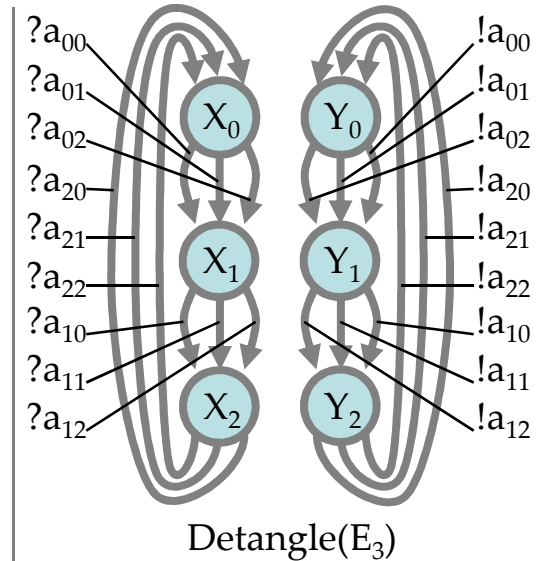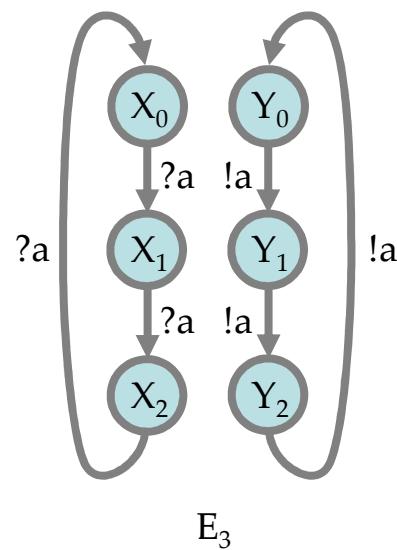
(closely related to
Pi(Ch(E_3)))

n² Scaling Problems

- E_n has 2n variables (nodes) and 2n terms (arcs).
- $Ch(E_n)$ has 2n species and n^2 reactions.

- The stoichiometric matrix has size $2n \cdot n^2 = 2n^3$.
- The ODEs have 2n variables and $2n(n+n) = 4n^2$ terms
  (number of variables times number of accretions plus depletions when sums are distributed)

E_3

$X_0 = ?a_{(r)};X_1$
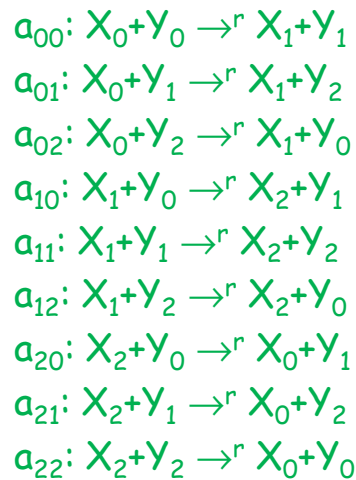$X_1 = ?a_{(r)};X_2$
$X_2 = ?a_{(r)};X_0$
$Y_0 = !a_{(r)};Y_1$
$Y_1 = !a_{(r)};Y_2$
$Y_2 = !a_{(r)};Y_0$

$Ch(E_3)$

$a_{00}: X_0+Y_0 \to^r X_1+Y_1$
$a_{01}: X_0+Y_1 \to^r X_1+Y_2$
$a_{02}: X_0+Y_2 \to^r X_1+Y_0$
$a_{10}: X_1+Y_0 \to^r X_2+Y_1$
$a_{11}: X_1+Y_1 \to^r X_2+Y_2$
$a_{12}: X_1+Y_2 \to^r X_2+Y_0$
$a_{20}: X_2+Y_0 \to^r X_0+Y_1$
$a_{21}: X_2+Y_1 \to^r X_0+Y_2$
$a_{22}: X_2+Y_2 \to^r X_0+Y_0$

StoichiometricMatrix($Ch(E_3)$)

	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}	a_{20}	a_{21}	a_{22}
X_0	-1	-1	-1				+1	+1	+1
X_1	+1	+1	+1	-1	-1	-1			
X_2				+1	+1	+1	-1	-1	-1
Y_0	-1		+1	-1		+1	-1		+1
Y_1	+1	-1		+1	-1		+1	-1	
Y_2		+1	-1		+1	-1		+1	-1

ODE(E_3)

$d[X_0]/dt = -r[X_0][Y_0] - r[X_0][Y_1] - r[X_0][Y_2] + r[X_2][Y_0] + r[X_2][Y_1] + r[X_2][Y_2]$
$d[X_1]/dt = -r[X_1][Y_0] - r[X_1][Y_1] - r[X_1][Y_2] + r[X_0][Y_0] + r[X_0][Y_1] + r[X_0][Y_2]$
$d[X_2]/dt = -r[X_2][Y_0] - r[X_2][Y_1] - r[X_2][Y_2] + r[X_1][Y_0] + r[X_1][Y_1] + r[X_1][Y_2]$
$d[Y_0]/dt = -r[X_0][Y_0] - r[X_1][Y_0] - r[X_2][Y_0] + r[X_0][Y_2] + r[X_1][Y_2] + r[X_2][Y_2]$
$d[Y_1]/dt = -r[X_0][Y_1] - r[X_1][Y_1] - r[X_2][Y_1] + r[X_0][Y_0] + r[X_1][Y_0] + r[X_2][Y_0]$
$d[Y_2]/dt = -r[X_0][Y_2] - r[X_1][Y_2] - r[X_2][Y_2] + r[X_0][Y_1] + r[X_1][Y_1] + r[X_2][Y_1]$

$=$
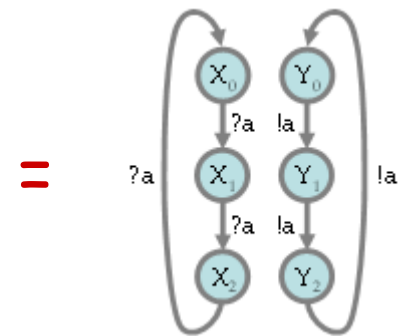
Conclusions

Conclusions

- **Compositional models**
  - Accurate (at the "appropriate" abstraction level).
  - Manageable (so we can scale them up by composition).
  - Executable (stochastic simulation).

- **Analysis techniques**
  - Mathematical techniques: Markov theory,
    Chemical Master Equation, and Rate Equation
  - Computing techniques: Abstraction and Refinement,
    Model Checking, Causality Analysis.

- **Many lines of extensions**
  - Parametric processes for model factorization
  - *Poly*automata for **Bio**-Chemistry: complexation and polymerization
  - Ultimately, rich process-algebra based modeling languages.

- **Quantitative techniques**
  - Important in the "real sciences".

Luca Cardelli